MỜI CÁC BẠN TÌM ĐỌC

1. GIÁO TRÌNH NGÔN NGỮ LẬP TRÌNH C/C++
2. GIÁO TRÌNH NGÔN NGỮ LẬP TRÌNH PASCAL
3. GIÁO TRÌNH CƠ SỞ DỮ LIỆU
4. GIÁO TRÌNH CƠ SỞ DỮ LIỆU PHÂN TÁN
5. GIÁO TRÌNH CƠ SỞ DỮ LIỆU: LÝ THUYẾT VÀ THỰC HÀNH
6. NHẬP MÔN PHÂN TÍCH THÔNG TIN CÓ BẢO MẬT
7. KỸ NGHỆ PHẦN MỀM
8. LÝ THUYẾT HỆ THỐNG VÀ ĐIỀU KHIỂN HỌC
9. SÁNG TẠO TRONG THUẬT TOÁN VÀ LẬP TRÌNH (3 TẬP)
10. KỸ THUẬT PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN HƯỚNG CẤU TRÚC

Giá: ...đ

TS. THÁI THANH TÙNG

GIÁO TRÌNH MẬT MÃ HỌC VÀ AN TOÀN THÔNG TIN

NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG

TS. THÁI THANH TÙNG

Giáo trình
MẬT MÃ HỌC & AN TOÀN THÔNG TIN

(CRYPTOGRAPHY AND SECURED INFORMATION)

NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG

TS. THÁI THANH TÙNG

Giáo trình

MẬT MÃ HỌC
&
HỆ THỐNG THÔNG TIN AN TOÀN

(CRYPTOGRAPHY AND SECURE INFORMATION SYSTEM)

NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG

GD 15 HM 11

LỜI GIỚI THIỆU

Với sự bùng nổ của Công nghệ thông tin vào cuối thế kỷ XX đầu thế kỷ XXI, nhân loại đang bước vào một thời đại mới: Thời đại của nền kinh tế thông tin toàn cầu hóa. Mọi hoạt động xã hội, chính trị, kinh tế trong thời đại mới hiện nay xét cho cùng, thực chất đều là những hoạt động thu thập, xử lý, lưu trữ và trao đổi thông tin. Trong bối cảnh đó An toàn và Bảo mật thông tin luôn là mối quan tâm hàng đầu trong mọi giao dịch xã hội, đặc biệt là giao dịch điện tử trên môi trường Internet, một môi trường mở, môi trường không được tin cậy.

TS. Thái Thanh Tùng dựa trên kinh nghiệm bản thân trong quá trình nhiều năm nghiên cứu, giảng dạy và hoạt động thực tế trong lĩnh vực an ninh mạng máy tính và bảo mật thông tin, đã tập hợp một số tài liệu cơ sở xuất bản trên thế giới trong những năm gần đây, đồng thời cập nhật những thành tựu mới nhất trong lĩnh vực nói trên để xây dựng nên cuốn sách này.

Cuốn sách được trình bày hợp lý với nội dung khá hoàn chỉnh, không những giúp cho người bắt đầu làm quen dễ tiếp thu những kiến thức cơ bản nhất của một lĩnh vực chuyên môn khó mà còn gợi mở những hướng ứng dụng thực tế phong phú cho những người muốn nghiên cứu sâu hơn.

Những phụ lục được sưu tầm chọn lọc đưa ra trong phần cuối cuốn sách có ý nghĩa bổ sung cho các phần trình bày chính và cũng là một sự hỗ trợ rất tốt về nguồn tư liệu cho những người muốn đi sâu nghiên cứu.

Giáo trình *Mật mã học và Hệ thống thông tin an toàn* của tác giả Thái Thanh Tùng đã được Ban Công nghệ *Viện Nghiên cứu và phát*

triển Tin học ứng dụng (*AIRDI*) thuộc Liên hiệp các Hội Khoa học và Kỹ thuật Việt Nam giới thiệu và *Hội đồng tư vấn ngành Công nghệ thông tin Viện Đại học Mở Hà Nội* đã chấp nhận sử dụng làm giáo trình chính thức để giảng dạy học phần *An ninh và Bảo mật thông tin* trong chương trình đào tạo Kỹ sư Công nghệ thông tin cũng như *Khoa Quốc tế Đại học Quốc gia Hà Nội* sử dụng trong chương trình đào tạo Cao học Quản lý Thông tin liên kết với *Đại học Lunghwa - Đài Loan*.

Xin trân trọng giới thiệu cùng bạn đọc!

Hà Nội, tháng 7 năm 2011

TS. TRƯƠNG TIẾN TÙNG

Trưởng Ban Công nghệ

Viện NC & PT Tin học Ứng dụng

LỜI MỞ ĐẦU

Con người luôn sống trong môi trường trao đổi thông tin hàng ngày, hàng giờ. Người thợ săn hú gọi bạn trong rừng thẳm, người đốc công niêm yết lệnh phân công trên bảng tin tức của công trường, người khách gửi đơn đặt hàng đến cửa hàng, con cái đi xa gọi điện thoại, gửi thư về báo tình hình cho bố mẹ,... tất cả những chuyện thường ngày đó đều chính là trao đổi thông tin.

Trong phần lớn trường hợp trao đổi thông tin giữa hai đối tác, người ta không hề muốn để thông tin bị lộ cho người thứ ba biết vì điều đó có thể gây ra những tổn thất cả về vật chất cũng như về tinh thần. Một báo cáo về một phát minh khoa học công nghệ mới, một bản phân tích tình hình giá cả hàng hóa ở một thị trường, một bộ hồ sơ dự thầu, nếu bị lộ ra trước khi đến tay người nhận thì thiệt hại kinh tế thật khó lường! Một vị nguyên soái gửi lệnh điều binh đến cho tướng lĩnh dưới quyền: chuyện gì sẽ xảy đến cho toàn quân nếu thông tin đó bị lộ cho kẻ địch biết?

Để bảo vệ bí mật cho thông tin của mình được gửi đi trong một môi trường "mở" tức là môi trường có thể có nhiều tác nhân tiếp cận ngoài hai đối tác trao đổi thông tin, người ta phải dùng mật mã tức là dùng những phương pháp biến đổi làm cho nguyên bản gốc của thông tin (*plaintext*) ở dạng thông thường ai cũng có thể hiểu được biến thành một dạng bí mật (*ciphertext*) mà chỉ có những người nắm được quy luật mới có thể biến đổi ngược lại thành dạng nguyên gốc ban đầu để đọc.

Mật mã học là khoa học nghiên cứu cơ sở lý thuyết và công nghệ để thực hiện việc xây dựng và sử dụng các hệ thống mật mã.

Mật mã học (*cryptography*) là một lĩnh vực liên quan đến các kỹ thuật ngôn ngữ học và toán học để đảm bảo an toàn thông tin, cụ thể là trong thông tin liên lạc. Quá trình mã hóa được sử dụng chủ yếu để đảm bảo tính bí mật của các thông tin quan trọng, chẳng hạn trong công tác tình báo, quân sự hay ngoại giao cũng như các bí mật về kinh tế, thương mại hay cả đến những thông tin cá nhân riêng tư.

Trong những năm gần đây, lĩnh vực hoạt động của mật mã hóa đã được mở rộng: mật mã hóa hiện đại cung cấp cơ chế cho nhiều hoạt động hơn là chỉ duy nhất việc giữ bí mật thông tin và còn có một loạt các ứng dụng quan trọng như: chứng thực khóa công khai, chữ ký số, thanh toán điện tử hay tiền điện tử. Ngay cả những người không có nhu cầu cao về tính bí mật và không có kiến thức về lập mật mã, giải mật mã cũng có thể sử dụng các công nghệ mật mã hóa, thông thường được thiết kế và tích hợp sẵn trong các cơ sở hạ tầng của công nghệ tính toán và liên lạc viễn thông.

Mật mã học là một ngành có lịch sử từ hàng nghìn năm nay. Trong phần lớn thời gian phát triển của mình (ngoại trừ mấy thập kỷ gần đây), lịch sử mật mã học chính là lịch sử của những phương pháp mật mã học cổ điển - các phương pháp mật mã hóa với bút và giấy, đôi khi có hỗ trợ từ những dụng cụ cơ khí đơn giản. Vào đầu thế kỷ XX, sự xuất hiện của các cơ cấu cơ khí và điện cơ, chẳng hạn như *máy Enigma*, đã cung cấp những cơ chế phức tạp và hiệu quả hơn cho mật mã hóa.

Sự ra đời và phát triển mạnh mẽ của ngành điện tử và máy tính trong những thập kỷ gần đây đã tạo điều kiện để mật mã học phát triển nhảy vọt lên một tầm cao mới.

Sự phát triển của mật mã học luôn đi kèm với sự phát triển của các kỹ thuật phá mã (hay còn gọi là *thám mã*). Các phát hiện và ứng dụng của các kỹ thuật phá mã trong một số trường hợp đã có ảnh hưởng đáng kể đến các sự kiện lịch sử. Một vài sự kiện đáng ghi nhớ bao gồm việc phát hiện ra bức điện Zimmermann đã khiến Hoa Kỳ tham gia Thế chiến II và việc phá mã thành công hệ thống mật mã của Đức quốc xã góp phần làm đẩy nhanh thời điểm kết thúc Thế chiến II.

Cho tới đầu thập kỷ 1970, các kỹ thuật liên quan tới mật mã học hầu như chỉ nằm trong tay các chính phủ. Hai sự kiện đã khiến cho mật mã học trở nên thích hợp cho mọi người, đó là: sự xuất hiện của tiêu chuẩn mật mã hóa dữ liệu DES (*Data Encryption Standard)* và sự ra đời của các kỹ thuật mật mã hóa khóa công khai.

Từ hơn mười năm trước, cứ vào tháng giêng hàng năm một số nhà nghiên cứu hàng đầu thế giới có một cuộc gặp gỡ trao đổi tại thung lũng Silicon được gọi là Hội thảo An ninh RSA – *RSA security Conference* (*John Kinyon*). Trong những năm đầu chỉ có một số ít nhà Toán học, Mật mã học, các Tư tưởng gia tiên phong trong những lĩnh vực liên quan đến an ninh dữ liệu cho máy tính điện tử và bảo mật thông tin trong giao dịch điện tử tham gia. Trong những năm cuối của thiên niên kỷ trước, vào thời kỳ bùng nổ của Công nghệ thông tin và Internet, vai trò quan trọng của các hội thảo an ninh điện tử đó ngày một nổi bật lên và hàng năm ngoài hội thảo an ninh RSA còn có hàng chục hội thảo an ninh thông tin điện tử và an ninh mạng khác được tiến hành, tập hợp sự tham dự và đóng góp của những tài năng kiệt xuất nhất trong kỷ nguyên công nghệ thông tin này.

Có thể khẳng định rằng, nếu không giải quyết được vấn đề an toàn dữ liệu cho máy tính điện tử, an ninh giao dịch điện tử (đặc biệt

là trên Internet) thì hầu như phần lớn thành quả của công nghệ thông tin, của mạng Internet đều trở thành vô nghĩa!

Do vậy, mọi kỹ sư, kỹ thuật viên, nhà nghiên cứu, người ứng dụng công nghệ thông tin đều cần được trang bị những kiến thức cơ bản tối thiểu về Mật mã học. Nhằm mục đích đó, tác giả đã sử dụng những tư liệu, giáo trình đã giảng dạy về mật mã học cho bậc đại học, cao học ngành công nghệ thông tin, toán tin ở Đại học Bách khoa Hà Nội, Viện Đại học Mở Hà Nội, tham khảo những công trình công bố quốc tế và trong nước trong vòng mười năm gần đây (xem tài liệu tham khảo) để biên soạn thành cuốn sách này. ***Giáo trình mật mã học và hệ thống thông tin an toàn*** là sự sắp xếp trình bày theo quan điểm của tác giả, có tham khảo nhiều tài liệu nhưng không dựa theo khuôn mẫu của một tư liệu nào cùng chuyên ngành đã công bố trước đây. Tác giả không dám hy vọng trình bày được thật chi tiết đầy đủ và đi sâu vào những vấn đề toán học rất phức tạp, mà chỉ mong đáp ứng phù hợp với nhu cầu của đông đảo sinh viên, kỹ sư, nhà nghiên cứu trong việc tìm hiểu một cách căn bản về một ngành học đang có hàng loạt ứng dụng quan trọng trong công nghệ thông tin và truyền thông hiện nay.

Nội dung giáo trình trình bày những khái niệm và định nghĩa chung về bảo mật thông tin, đi sâu phân tích 2 loại mã hóa: mã khóa bí mật cùng các giao thức, thuật toán trao đổi khóa mã và mã bất đối xứng hay mã khóa công khai và khóa riêng với những ứng dụng cụ thể của nó. Bên cạnh đó, nội dung giáo trình giới thiệu đến một vấn đề rất có ý nghĩa hiện nay trong các giao dịch thương mại điện tử, ngân hàng trực tuyến đó là: *Chữ ký điện tử, chữ ký số và vấn đề phân phối khóa công khai với các hệ thống hạ tầng cơ sở khóa công khai PKI và chuẩn X509 cũng như hệ thống mạng lưới tin cậy và giao thức PGP*. Đặc biệt phần cuối giới thiệu các giao thức và chuẩn mã

hóa thông dụng nhất trên Internet trong các dịch vụ bảo mật thư điện tử như *S/MIME*, những giao thức và chuẩn mã hóa sử dụng để bảo đảm an toàn thông tin đặc biệt quan trọng trong thương mại điện tử, ngân hàng điện tử, như *SSL/TLS và HTTPS, FTPS, SET, SSH, IPsec*… Ở cuối mỗi phần lý thuyết, giáo trình cung cấp một danh mục các phần mềm ứng dụng thương mại và phi thương mại để người đọc tiện tra cứu, sử dụng.

Giáo trình được xuất bản lần đầu sẽ khó tránh khỏi những thiếu sót. Rất mong nhận được ý kiến nhận xét, góp ý của bạn đọc để giáo trình ngày càng được hoàn thiện hơn trong lần tái bản sau.

Xin chân thành cảm ơn các bạn đồng nghiệp ở Khoa Công nghệ Thông tin - Viện Đại học Mở Hà Nội đã góp ý cho tác giả trong việc biên soạn giáo trình này.

Hà Nội, tháng 7 năm 2011

Tác giả

1

TỔNG QUAN VỀ BẢO MẬT THÔNG TIN VÀ LÝ THUYẾT MÃ HÓA

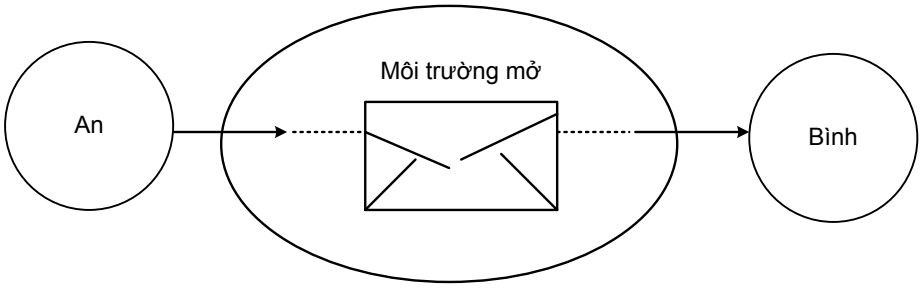
1.1. NHU CẦU BẢO MẬT THÔNG TIN GIAO DỊCH TRONG MÔI TRƯỜNG MỞ

Trong toàn bộ cuốn sách này chúng ta sẽ quy ước xem xét các giao dịch giữa hai đối tác: An (A) là người gửi (phát) thông tin và Bình (B) là người nhận (thu) thông tin. Ngoài hai đối tác nói trên chúng ta cũng giả thiết rằng tồn tại một kẻ thứ ba là Công (C), C luôn tìm cách xâm nhập những thông tin trao đổi giữa A và B để nghe lén (trộm thông tin) hoặc để thay đổi làm sai lệch các thông tin được trao đổi giữa A và B nhằm một mục đích nào đó.

Giả sử An có một câu chuyện riêng tư bí mật cần nói với Bình. Rõ ràng lý tưởng nhất là hai người có thể kéo nhau vào một căn phòng cửa đóng kín (tường cách âm càng tốt) và thì thào với nhau: mọi điều trao đổi chỉ có hai người biết, không lọt vào tai bất kỳ một người thứ ba nào. Môi trường giao dịch đó là một *môi trường đóng* (theo nghĩa là ngoài hai đối tác giao dịch, không có sự xâm nhập của bất kỳ một người thứ ba nào), môi trường đóng là một *môi trường tin cậy*.

Tuy nhiên trong thực tế, người ta thường phải tiến hành giao dịch trong những môi trường không đóng, tức là *môi trường mở* (*open surrounding*). Chẳng hạn, vì gấp quá, không tìm ra chỗ kín

đáo, An phải đứng ngay đầu đường nói to lên với Bình đang đứng ở cuối đường, câu chuyện hiển nhiên lọt vào tai của nhiều người khác. Hoặc An đang ở Hà Nội phải gọi điện thoại hay gửi thư cho Bình ở TP. Hồ Chí Minh, không thể đảm bảo là nội dung cuộc nói chuyện điện thoại hoặc nội dung lá thư không bị người thứ ba nào đó nắm bắt được. Môi trường mở nói chung là *môi trường không tin cậy*.



Hình 1.1: Môi trường mở trong trao đổi thông tin

1.2. NHỮNG NGUYÊN LÝ CỦA BẢO MẬT THÔNG TIN

Các giao dịch điện tử nói chung là giao dịch trong môi trường mở, giao dịch trên Internet, giao dịch xuyên quốc gia. Trong các quá trình trao đổi thông tin đó các đối tác thường là không "mặt đối mặt" để có thể nhận diện ra nhau. Vì thế rất khó để có thể thực hiện được những yêu cầu sau đây của việc trao đổi thông tin được xem là những nguyên lý cơ bản của vấn đề bảo mật thông tin:

1. Tính bí mật/riêng tư.
2. Tính toàn vẹn.
3. Tính xác thực.
4. Tính không thể chối bỏ.
5. Tính nhận dạng.

Thêm vào đó, tốc độ thực hiện truyền tin (nhanh chóng) cũng là một yêu cầu cần chú ý. Ta sẽ lần lượt xét qua các yêu cầu đã kể trên.

1.2.1. Nguyên lý 1: Nguyên lý bí mật/riêng tư (Confidentiality/Privacy)

Giả sử A gửi một "vật mang tin" đến cho B. Nguyên lý đầu tiên của lý thuyết bảo mật là phải đảm bảo tính bí mật và tính riêng tư cho quá trình truyền tin. Điều này có nghĩa là việc truyền tin phải đảm bảo rằng chỉ có hai đối tác A và B khi tiếp cận vật mang tin mới nắm bắt được nội dung thông tin được truyền. Trong quá trình truyền tin, nếu có kẻ thứ ba C (vì một nguyên nhân nào đó) có thể tiếp cận được vật mang tin thì phải đảm bảo rằng kẻ đó vẫn không thể nắm bắt được, không thể hiểu được nội dung "thực sự" của thông tin chứa trong vật mang tin đó.

1.2.2. Nguyên lý 2: Nguyên lý toàn vẹn (Integrity)

Trong quá trình truyền tin, có thể vì lý do khách quan của môi trường, nhất là do sự xâm nhập phá hoại của kẻ thứ ba, nội dung của thông tin ban đầu chứa trong vật mang tin có thể bị mất mát hay bị thay đổi. Nguyên lý này không yêu cầu đến mức phải đảm bảo rằng thông tin không bị thay đổi trong quá trình truyền tin, nhưng phải đảm bảo được là mỗi khi thông tin bị thay đổi thì người nhận (và tất nhiên là cả người gửi) đều phát hiện được. Chẳng hạn vật mang tin của A gửi cho B trên đường truyền tạm thời lọt vào tay người thứ ba C. C tuy không thể hiểu được nội dung thông tin (do quá trình truyền tin đã thực hiện nguyên lý 1) nhưng vẫn có thể tác động vào vật mang tin để làm thay đổi thông tin nó mang; khi nhận được vật mang tin (đã bị làm thay đổi) B lập tức nhận biết rằng nó đã bị làm thay đổi.

1.2.3. Nguyên lý 3: Nguyên lý xác thực (Authentication)

Nguyên lý 3 của bảo mật thông tin yêu cầu là trong một quá trình truyền tin, người nhận tin (và có khi cả người gửi tin nữa) có

biện pháp để chứng minh với đối tác rằng "*họ chính là họ*" chứ không phải là một người thứ ba nào khác. Chẳng hạn khi bạn nhận một lá thư bảo đảm tại Bưu điện thì bạn phải có cách nào chứng minh được rằng bạn chính là người có quyền nhận lá thư đó, bằng cách xuất trình chứng minh nhân dân hoặc một giấy giới thiệu có giá trị nào đó. Sự xác thực này có thể là xác thực một chiều (*one-way authentication*): người nhận phải xác thực mình với người gửi, nhưng cũng có những trường hợp đòi hỏi xác thực hai chiều (*mutual authentication*): người nhận với người gửi và ngược lại. Chẳng hạn khi A là khách hàng gửi tin báo cho B là chủ nhà hàng chuẩn bị cho mình một bữa tiệc, A phải xác thực được rằng người nhận tin của mình đúng là B (người có trách nhiệm của nhà hàng) chứ không phải là một nhân viên nào đó có thể vô trách nhiệm, quên lãng làm nhỡ nhàng cho khách của mình. Mặt khác khi B nhận tin cũng phải xác thực được đúng là đơn đặt hàng của A chứ không phải do một kẻ phá rối nào đó mạo danh làm cho mình bị ế bữa tiệc đã chuẩn bị.

1.2.4. Nguyên lý 4: Nguyên lý không chối bỏ (Non repudition)

Nguyên lý này đòi hỏi rằng khi quá trình truyền tin kết thúc, A đã gửi cho B một thông tin và B đã nhận thông tin thì A không thể chối bỏ rằng thông tin đó không do mình gửi (hoặc mình không gửi tin) mặt khác B cũng không thể chối bỏ rằng mình chưa nhận được. Cũng trong ví dụ về việc đặt tiệc nói trên, nếu A đã đặt tiệc nhưng không đến ăn thì không thể chối là tin đặt tiệc không do mình gửi, ngược lại khi khách khứa đến mà B quên chuẩn bị thì B cũng không thể chối là do mình chưa nhận được đơn đặt hàng của A.

1.2.5. Nguyên lý 5: Nguyên lý nhận dạng (Identification)

Giả sử một hệ thống tài nguyên thông tin chung có nhiều người sử dụng (users) với những mức độ quyền hạn khác nhau. Nguyên lý 5 của bảo mật thông tin yêu cầu phải có biện pháp để hệ thống có thể

nhận dạng được các người sử dụng với quyền hạn kèm theo của họ. Chẳng hạn trong một thư viện có nhiều kho sách chứa các loại tài liệu thông thường và tài liệu mật. Người đọc chia làm nhiều loại, có loại chỉ được đọc sách thông thường tại chỗ, có loại được đọc tài liệu mật, có loại lại được mượn về nhà. Người vào thư viện phải xuất trình thẻ, có các loại thẻ khác nhau: Căn cứ vào thẻ, người thủ thư nhận dạng được ra người đó có phải là người có quyền sử dụng thư viện không và có quyền sử dụng theo dạng nào.

Trong vấn đề bảo mật còn có một điều cần lưu ý: đó là "sự tin tưởng". Khi chia sẻ một bí mật cho một người, bạn phải tin tưởng vào khả năng bảo vệ bí mật của người đó. Nhưng một điều khó khăn ở đây là: "tin tưởng" là một phạm trù có tính tâm lý, xã hội không có các đặc trưng của một loại quan hệ toán học nào:

- *Tính không phản xạ:* Một người có luôn luôn tin tưởng vào chính mình không? (Điều này chưa chắc chắn đối với tất cả mọi người và trong tất cả mọi trường hợp!)
- *Tính không đối xứng:* A tin tưởng vào B nhưng liệu B có tin tưởng vào A không? (Chưa chắc!)
- *Tính không bắc cầu:* A tin tưởng B, B tin tưởng C, nhưng không có gì đảm bảo (trong rất nhiều trường hợp) là A tin tưởng vào C.

Chính vì vậy, trong các vấn đề bảo mật nhiều khi chúng ta không thể hoàn toàn dùng các phương pháp suy luận logic thông thường mà phải chú ý đến việc tuân thủ các nguyên lý bảo mật thông tin.

1.3. KHÁI NIỆM VÀ THUẬT NGỮ

Trong mục này chúng ta thống nhất với nhau một số thuật ngữ thường dùng sau này.

Thông điệp (*message*) là một thực thể vật lý mang thông tin cần trao đổi. Lá thư, điện tín (*telegraph)*, E-mail là thông điệp dạng văn bản (*text*). Câu chuyện qua điện thoại, bài nói trên đài phát thanh, phát biểu trong một cuộc họp... là những thông điệp dạng âm thanh (*sound*). Các album ảnh, các bức tranh... là những thông điệp dạng ảnh (*picture*), còn một bộ phim câm, một videoclip không có tiếng nói là những thông điệp dạng hình ảnh động (*animation*). Các thông điệp bao gồm cả bốn dạng trên là những thông điệp đa phương tiện (*multimedia*) chẳng hạn như một cuộn băng video, một chương trình truyền hình... đều là những thông điệp multimedia. Trong giao dịch điện tử, mọi thông điệp dù bất cứ ở dạng nào cũng đều được số hóa, tức là chuyển đổi thành những dãy bit, những dãy số nhị phân chỉ gồm hai con số 0 và 1. Vì vậy có thể nói rằng: **Mọi thông điệp điện tử đều là những dãy con số dạng nhị phân**. Nhưng mỗi con số dạng nhị phân lại đều có thể chuyển trở lại thành dạng thập phân. Cho nên người ta cũng có thể dùng một con số thập phân để biểu diễn một thông điệp. Chẳng hạn khi có thông điệp đã số hóa thành số nhị phân là: **1111011** ta cũng có thể nói rằng thông điệp đó là số thập phân **123**. Vì vậy trong giao dịch điện tử hiện đại, khi xem xét việc xử lý các thông điệp điện tử chúng ta hiểu rằng đấy là việc xử lý các thông điệp số hóa.

Plain text/message: là thông điệp, dữ liệu "*gốc*" dạng "*tường minh*" dạng ban đầu của người phát hành thông điệp tạo ra, mọi người bình thường trong cùng môi trường xã hội với người tạo ra và người được gửi thông điệp (và cả những người thứ ba vì lý do nào đó có cơ hội tiếp cận được thông điệp đó) đều có thể hiểu được nội dung. Chẳng hạn trong xã hội có nhiều người biết tiếng Việt, An viết một lá thư bằng tiếng Việt gửi cho Bình: lá thư là một plaintext vì nếu nhận được lá thư thì không những chỉ có Bình hiểu được nội dung mà bất kỳ người nào biết tiếng Việt có được lá thư cũng hiểu ngay nội dung lá thư đó.

Cipher text/message: là thông điệp dữ liệu đã *biến đổi theo một quy tắc nào đó* thành một dạng khác (dạng "*ẩn tàng*") mà chỉ những người nào nắm bắt được *quy tắc biến đổi ngược* trở lại thành plaintext thì mới hiểu được nội dung thông điệp. Chẳng hạn trong một môi trường, ngoài An và Bình không có người nào khác biết tiếng Anh. Sau khi An viết một bức thư bằng tiếng Việt (plaintext) trước khi gửi cho Bình đã dịch ra tiếng Anh, khi lá thư đến tay Bình, vì Bình cũng biết tiếng Anh nên dễ dàng dịch ngược lại để hiểu nội dung, còn nếu bản dịch của lá thư ra tiếng Anh rơi vào tay Công, do Công (cũng như mọi người xung quanh) không biết tiếng Anh nên không thể hiểu được nội dung. Bản dịch lá thư ra tiếng Anh trong trường hợp này được xem là một ciphertext.

Cipher (hay cypher): là thuật toán dùng để chỉ quy tắc để thực hiện việc biến đổi thông điệp dạng tường minh (plaintext) thành thông điệp dạng ẩn tàng (ciphertext), quá trình này gọi là mã hóa và cũng để chỉ quá trình biến đổi ngược từ ciphertext trở lại thành plaintext, quá trình này gọi là giải mã. Trong khuôn khổ cuốn sách này ta đều gọi các quy tắc đó là những thuật toán.

Encrypt (encipher, encryption: mã hóa): đó là quá trình biến đổi thông tin từ dạng ban đầu (dạng tường minh) thành dạng ẩn tàng, với mục đích giữ bí mật thông tin đó.

Decrypt (decipher, decryption: giải mã): đó là quá trình ngược lại với mã hóa, khôi phục lại những thông tin dạng ban đầu từ thông tin ở dạng đã được mã hóa.

Cryptosystem (Cryptographic system: Hệ thống mã hóa thông tin): có thể là các phần mềm như PGP, Ax-Crypt, Truecrypt... các giao thức như SSL, IPsec dùng trong Internet... hay đơn giản là một thuật toán như DEA.

Chìa khóa (Key): chính là thông tin dùng cho quy trình mã hóa và giải mã. Password (mật khẩu) là một hay dãy ký tự, ký hiệu, tín

hiệu mà người dùng được hệ thống bảo mật cấp để xác nhận cấp quyền được phép truy cập hoặc can thiệp ở một mức độ quy định (xem, nghe, sửa, xóa...) vào một khu vực lưu trữ thông tin nào đó. Trong thực tế, mật khẩu do người dùng tạo ra thường không đủ độ an toàn để được dùng trực tiếp trong thuật toán.

Vì vậy, trong bất cứ hệ thống mã hóa dữ liệu nghiêm túc nào cũng phải có bước chuyển đổi mật khẩu ban đầu thành chìa khóa có độ an toàn thích hợp, thường gọi là công đoạn tạo chìa khóa. Bước tạo chìa khóa này thường được gọi là *key derivation, key stretching* hay *key initialization*.

Key Derivation Function (Hàm tạo khóa): thường sử dụng một hàm băm (*hash function*) (sẽ giải thích rõ hơn ở phần sau) được thiết kế sao cho chìa khóa an toàn hơn đối với các kiểu tấn công thám mã. Hàm này được thực hiện lại nhiều lần trên mật khẩu ban đầu cùng với một con số ngẫu nhiên để tạo ra một chìa khóa có độ an toàn cao hơn. Con số ngẫu nhiên này gọi là *salt*, còn số lần lặp lại là *iteration*. Ví dụ một mật khẩu là "pandoras B0x", cùng với salt là "230391827", đi qua hàm hash SHA-1 1000 lần cho kết quả là một chìa khóa có độ dài 160 bit (thể hiện dưới dạng số thập lục phân: hệ đếm cơ số 16) như sau:

3BD454A72E0E7CD6959DE0580E3C19F51601C359

Keylength (Keysize): Độ dài (hay kích thước) của chìa khóa. Ta nói một chìa khóa có độ dài 128 bit có nghĩa chìa khóa đó là một số nhị phân có độ dài 128 chữ số. Ta sẽ thấy rằng một thuật toán có chìa khóa càng dài thì càng có nhiều khả năng chống lại các kiểu tấn công. (Bạn có thể so sánh như số viên bi trong một ổ khóa bi thường dùng: số bi càng nhiều thì ổ khóa càng an toàn).

Xem xét một ví dụ sau đây. Một thợ khóa tài giỏi tạo ra một ổ khóa kiểu tổ hợp (*combination lock:* loại ổ khóa được khóa (hay mở)

bằng cách xoay một số lần theo chiều thuận và một số lần theo chiều ngược kim đồng hồ đến những con số nào đó, như các ổ khóa két sắt) và hướng dẫn cách sử dụng cho khách hàng. An và Bình mỗi người mua một ổ khóa kiểu đó mang về và mỗi người đặt một kiểu tổ hợp khác nhau cho mình. Lúc đó thì tuy là dùng chung một loại khóa nhưng An và Bình không thể mở được ổ khóa của nhau, kể cả người thợ khóa cũng không mở được khóa của hai người! Ổ khóa kiểu tổ hợp là một thuật toán mã hóa và giải mã. Cách chọn tổ hợp của An hay Bình là những khóa (*key*) khác nhau. Số lần quay ổ mà An hay Bình chọn để khóa (và mở) chính là độ dài của khóa. Nếu An không biết khóa của Bình đã đặt mà muốn "phá khóa" thì thông thường phải "dò thử" mọi khả năng có thể có của các tổ hợp, bằng không thì phải ... vác búa ra mà đập vỡ ổ khóa! Kiểu phá khóa bằng cách dò thử tất cả mọi khả năng như vậy gọi là "*tấn công bạo lực: brute force attack*", tất nhiên tấn công kiểu đó bao giờ cũng thành công (nghĩa là phá được ổ khóa) nhưng rõ ràng phương pháp đó tốn rất nhiều thời gian.

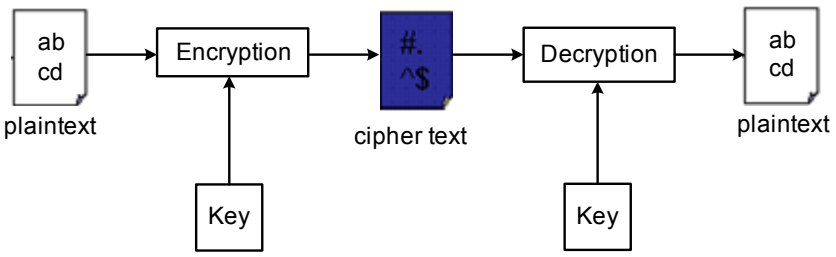
Độ dài khóa càng lớn (tức là số lần mà người chủ khóa quy định phải quay để khóa hoặc mở) thì việc tấn công bạo lực càng mất nhiều thời gian. Người ta đánh giá một ổ khóa là đủ an toàn trong một thời gian T nếu như khả năng tấn công bạo lực phải mất thời gian gấp nhiều lần T. Chẳng hạn một người thường vắng nhà không quá 7 ngày, nếu ổ khóa chỉ có thể phá được bằng tấn công bạo lực trong suốt một tuần thì ổ khóa được xem là an toàn. Nhưng nếu người đó đi xa 1 tháng thì sử dụng ổ khóa đó là không an toàn nữa!

Một chuyên gia phá khóa có thể có những phương pháp dò tìm khác mà thời gian phá khóa rất ít so với kiểu tấn công bạo lực. Như vậy muốn đánh giá mức độ an toàn của một ổ khóa ta cần phải xem xét *mọi khả năng phá khóa* có thể có chứ không phải chỉ đánh giá qua thời gian tấn công bạo lực.

1.4. MẬT MÃ HỌC

1.4.1. Mật mã học (cryptography) là gì?

Người ta gọi mật mã học là một khoa học nghiên cứu nghệ thuật nhằm che giấu thông tin, bằng cách mã hóa (*encryption*) tức là biến đổi "thông tin gốc" dạng tường minh (plaintext) thành "thông tin mã hóa" dạng ẩn tàng (cipher text) bằng cách sử dụng một *khóa mã* (thuật toán mã hóa) nào đó. Chỉ có những người giữ *chìa khóa* (*key*) bí mật mới có thể giải mã (*decryption*) thông tin dạng ẩn tàng trở lại thành dạng thông tin có dạng tường minh.



Hình 1.2: Sơ đồ mã hóa và giải mã

Thông tin ẩn tàng đôi khi vẫn bị khám phá mà không cần biết khóa bí mật: việc đó gọi là *bẻ khóa*. Ngành học nghiên cứu về việc *bẻ khóa* (*attack/crack/hack*) này còn gọi là *cryptanalysis*. Như đã nói ở ví dụ trên, trong các phương pháp tấn công thám mã ta gọi tấn công bạo lực - *brute-force attack* (*exhaustive key search*): là phương pháp tấn công bằng cách thử tất cả những khả năng chìa khóa có thể có. Đây là phương pháp tấn công thô sơ nhất và cũng khó khăn nhất. Theo lý thuyết, tất cả các thuật toán hiện đại đều có thể bị đánh bại bởi tấn công bạo lực nhưng trong thực tiễn việc này chỉ có thể thực hiện được trong thời gian rất dài nên thực tế là không khả thi. Vì thế có thể coi một thuật toán là an toàn nếu như không còn cách nào khác để tấn công nó ngoài cách sử dụng brute-force attack. Để chống lại tấn công này, chìa khóa bí mật được thay đổi một cách thường xuyên hơn.

Trong lý thuyết mật mã, người ta nghiên cứu đồng thời các thuật toán lập mã và vấn đề thám mã được dùng để đánh giá mức độ an toàn và khả năng bảo mật thông tin của mỗi thuật toán mã hóa.

1.4.2. Mật mã học trong lịch sử

Có thể xem là lịch sử mật mã học bắt nguồn từ người Ai Cập vào khoảng những năm 2000 trước Công nguyên khi họ dùng những ký hiệu tượng hình khó hiểu để trang trí trên các ngôi mộ nhằm bí mật ghi lại tiểu sử và những chiến tích, công lao của người đã khuất. Trong một thời gian dài hàng thế kỷ một trong những loại công trình nghiên cứu thu hút rất nhiều nhà khoa học trên thế giới là các nghiên cứu giải mã những "dấu tích bí mật" trên các ngôi mộ cổ Ai Cập, nhờ đó mà ta hiểu biết được khá nhiều về lịch sử, phong tục, tập quán sinh hoạt của đất nước Ai Cập cổ huyền bí.

Người Hebrew (Do Thái cổ) đã sáng tạo một thuật toán mã hóa đơn giản và hiệu quả gọi là thuật toán *atbash* mà chìa khóa mã hóa và giải mã là một sự thay thế (*substitution*) trong bảng chữ cái. Giả sử dùng chìa khóa mã hóa là bảng hoán vị:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

↓

Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

Khi đó chẳng hạn từ gốc (plaintext): **JERUSALEM** sẽ được mã hóa thành từ mã (ciphertext): **QVIFHZOVN**. Nếu người nhận tin có chìa khóa thì việc biến đổi QVIFHZOVN trở lại thành JERUSALEM là điều hoàn toàn đơn giản, nhưng nếu không có chìa khóa thì quả là khó khăn, người nhận được thông điệp không thể nào hiểu nổi QVIFHZOVN có nghĩa là gì cả! Cho dù biết rằng quy luật mã hóa chỉ là một sự thay thế của 25 chữ cái nhưng nếu tấn công bạo lực thì phải thử lần lượt hết mọi khả năng tạo chìa khóa, tức là phải thử 25! khả năng (tất nhiên về sau người ta có rất nhiều biện pháp để giảm

bớt khả năng dò tìm, chẳng hạn nếu plaintext có độ dài khá lớn thì có thể sử dụng dò tìm theo tần suất xuất hiện của các ký tự).

Thuật toán mã hóa bằng thay thế này chỉ dùng một ký tự (chữ cái) thay thế cho một ký tự nên được gọi là thuật toán mã hóa thay thế đơn (*monoalphabetic substitution*). Người ta cũng có thể tạo những thuật toán mã hóa thay thế khối (*multiple alphabetic substitution*) nếu thay vì thay thế từng ký tự ta thay thế một dãy ký tự gốc bởi một dãy ký tự mã hóa: thuật toán này cho ta nhiều khả năng tạo khóa hơn nên khả năng bị tấn công lại càng giảm xuống.

Vào khoảng năm 400 trước CN, người Sparte sử dụng một dụng cụ gọi là gậy mật mã. Các đối tác viết thư lên một hàng ngang của mảnh giấy dài cuốn quanh một cây gậy có đường kính và độ dài quy ước với nhau trước rồi tháo ra và điền vào các ô trống những ký tự bất kỳ. Đối tác nhận thư phải có một cây gậy giống hệt, cùng đường kính và độ dài, lại quấn mảnh giấy vào gậy và "giải mã" được. Nếu không hiểu quy luật và không có cây gậy như thế thì không thể nào đọc hiểu những ký tự nối đuôi nhau một cách "vô nghĩa" trên mảnh giấy.

Về thời Trung Cổ, hoàng đế La Mã nổi tiếng là Julius Caesar tạo một công cụ lập mã rất đơn giản cho thuật toán gọi là "mã vòng" (*cyclic code*) tương tự như thuật toán atbash của người Hebrew nhưng đây không phải là một sự thay thế bất kỳ mà là một sự thay thế theo hoán vị vòng quanh. Caesar dùng hai vành tròn đồng tâm, trên cả hai vành đều ghi bảng chữ cái La-tinh, vành trong ứng với plaintext còn vành ngoài ứng với ciphertext. Chìa khóa mã hóa là phép xoay vành tròn bên ngoài một số bước, do đó các chữ cái thay đổi đi. Chẳng hạn nếu chìa khóa là +3 tức là xoay theo chiều thuận +3 ô thì các chữ cái A, B, C…X, Y, Z trong plaintext sẽ chuyển đến D, E, F …A, B, C trong ciphertext, từ HANOI trong plaintext được mã hóa thành từ KDQRL trong ciphertext. Người nhận sẽ giải mã bằng cách xoay ngược vành chữ ngoài -3 ô thì tìm lại được plaintext.

Ngày nay, các phương pháp mã hóa và lập mã đó xem ra quá đơn giản nên không còn được dùng trong các vấn đề bảo mật thông tin quan trọng, tuy nhiên cũng còn giá trị cho một số người khi muốn dùng để bảo mật những ghi chép cá nhân thông thường của mình và ý tưởng của chúng vẫn còn được sử dụng trong một số công cụ lập mã hiện đại. Mật mã học được phát triển mạnh ở châu Âu và mãi đến khoảng năm 1800 chủ yếu vẫn chỉ được sử dụng nhiều trong việc bảo mật các thông điệp quân sự. Chính nguyên lý mã vòng của Caesar là ý tưởng cho việc phát triển một thiết bị mã hóa nổi tiếng nhất trong lịch sử: *máy mã hóa Enigma* của người Đức dùng trong Đại chiến thế giới lần thứ hai. Enigma có 3 ổ quay, mỗi ký tự trong plaintext khi đưa vào sẽ được thay thế 3 lần theo những quy luật định sẵn khác nhau cho nên quá trình thám mã rất khó khăn.

Về sau một nhóm các nhà mật mã học Ba Lan đã bẻ khóa được thuật toán lập mã của Enigma và cung cấp cho người Anh mọi thông tin quân sự của Đức: người ta đánh giá rằng thành công của việc phá khóa đó đã rút ngắn thời gian kéo dài của Thế chiến II bớt được 2 năm. Sau khi Thế chiến II kết thúc, bí mật của Enigma được công bố và ngày nay một máy Enigma còn được triển lãm tại Viện Smithsonian, Washington D.C, Hoa Kỳ.

William Frederick Friedman (1891 – 1989)

Năm 1920, William Frederic Friedman công bố tác phẩm *The Index of Coincidence and Its Applications in Cryptography* (*Chỉ số trùng hợp và ứng dụng của nó vào Mật mã học*). Ông được xem là "cha đẻ của Mật mã học hiện đại".

1.4.3. Phân loại các thuật toán mã hóa

Ngày nay người ta phân biệt ra hai nhóm thuật toán mã hóa chính là: Các thuật toán mã hóa cổ điển và các thuật toán hiện đại.

- Các thuật toán cổ điển: (những thuật toán này ngày nay đôi khi vẫn còn được dùng chẳng hạn trong trò chơi tìm mật thư) gồm:

+ **Thuật toán thay thế** (*Substitution*) là thuật toán mã hóa trong đó từng ký tự (hoặc từng nhóm ký tự) của plaintext được thay thế bằng một (hay một nhóm) ký tự khác. Thuật toán atbash của người Hebrew hay thuật toán vòng của Caesar đều là các thuật toán thay thế. Chính ý tưởng của mã vòng Caesar đã được ứng dụng trong máy Enigma.

+ **Thuật toán chuyển vị** (*Transposition*) là thuật toán mã hóa trong đó các ký tự trong văn bản ban đầu chỉ thay đổi vị trí cho nhau còn bản thân các ký tự không hề bị biến đổi.

Xét một ví dụ về thuật toán hoán vị. Trong thuật toán này chúng ta ngắt thông điệp gốc thành từng nhóm 4 ký tự đánh số trong từng nhóm từ 1 đến 4. Chìa khóa ở đây là một hoán vị bất kỳ của 1234 gán cho mỗi nhóm:

HAI PHONG	**Plaintext**
H A I P H O N G	Ngắt đoạn từng nhóm 4 ký tự
1 2 3 4 1 2 3 4	Thứ tự tự nhiên trong mỗi nhóm
2 4 1 3 3 1 4 2	Khóa mã (chọn hoán vị tùy ý)
A P H I N H G O	**Ciphertext**

Các thuật toán hiện đại:

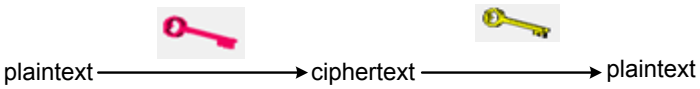
Có nhiều cách phân loại các thuật toán mã hóa hiện đại hiện đang sử dụng. Trong cuốn sách này ta sẽ phân biệt theo số chìa khóa sử dụng trong một thuật toán và như vậy có 3 loại sau đây:

a. Mã hóa đối xứng hay khóa bí mật SKC (*Secret Key Cryptography*): Chỉ dùng một chìa khóa cho cả mã hóa và giải mã (biến đổi theo hai chiều ngược nhau)

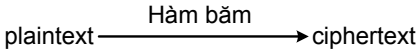
b. Mã hóa bất đối xứng hay khóa công khai và khóa riêng PKC (*Public and Private Keys Cryptography*): Sử dụng hai khóa riêng biệt: một khóa để mã hóa (khóa công khai: *public key*) và một khóa khác để giải mã (khóa riêng: *private key*).

c. Hàm băm (*Hash function*): Mã hóa một chiều (*one-way cryptography*) dùng một biến đổi toán học để "mã hóa" thông tin gốc thành một dạng không biến đổi ngược được: không có chìa khóa vì từ ciphertext không tìm ngược lại được plaintext!

a. Mã hóa khóa bí mật (đối xứng). SKC sử dụng một khóa cho cả mã hóa và giải mã.



b. Mã hóa khóa công khai (bất đối xứng). PKC sử dụng hai khóa, một khóa để mã hóa và khóa còn lại để giải mã.



c. Hàm băm (mã hóa một chiều). Hàm băm không có chìa khóa do plaintext không tìm ngược lại được ciphertext.

Hình 1.3: Khóa đối xứng, khóa bất đối xứng và hàm băm

Trong những chương sau chúng ta sẽ đi vào lần lượt nghiên cứu về các thuật toán lập mã và giải mã cho các loại mã đối xứng, mã bất đối xứng, ưu điểm và nhược điểm của chúng và khả năng ứng dụng của chúng trong việc truyền các thông điệp điện tử.

Sau đây chúng ta tham khảo ví dụ về thách thức bảo mật thời kỹ thuật số.

Việc phát tán thông tin mật của Bộ Quốc phòng Mỹ trên Wikileaks đang đặt ra thách thức an ninh trong thời đại kỹ thuật số.

Các nhà phân tích cho rằng ngày 26/7/2010 dữ liệu bị đánh cắp có dung lượng tính bằng gigabytes có thể được chia sẻ chỉ bằng một lần cái nhấp chuột.

"Tôi nghĩ về việc này trong mối liên hệ với Tài liệu Lầu Năm Góc", James Lewis, một chuyên gia mạng, tại Trung tâm Chiến lược và Nghiên cứu Quốc tế (CSIS), so sánh với sự cố năm 1971 khi dữ liệu trong hồ sơ Cuộc chiến Việt Nam của Lầu Năm Góc bị rò rỉ.

"Sự khác biệt với Tài liệu Lầu Năm Góc là ở chỗ Daniel Ellsberg lấy nhiều tài liệu ở dạng in trên giấy và đưa cho một phóng viên", ông Lewis nói.

"Nay người ta có thể lấy nhiều tài liệu hơn nhiều và phát tán cho toàn thế giới."

Wikileaks đã không xác định nguồn tài liệu mật nhưng mối nghi ngờ hiện đang nhắm tới Bradley Manning, một nhà phân tích tình báo quân đội Mỹ đang bị giam tại một nhà tù quân sự ở Kuwait.

Julian Assange, một nhà báo và là chủ trang Wikileaks cho báo chí Anh hay trong chuyến đến Luân Đôn rằng ông còn đang nắm trong tay hàng nghìn tư liệu như vụ vừa qua nhưng chưa tung ra.

Lầu Năm Góc tin tưởng nhân viên của mình, đó là điều tốt, nhưng không đủ.

James Lewis, Trung tâm Chiến lược và Nghiên cứu Quốc tế

Riêng Manning đã bị bắt vào tháng Năm sau khi Wikileaks phát đoạn băng video vụ một chiếc trực thăng Apache của Mỹ tại I-rắc tấn công và có dân thường chết trong vụ oanh tạc này.

Ông ta đã bị buộc tội cung cấp thông tin quốc phòng cho một nguồn trái phép.

Bộ Quốc phòng Mỹ trong tháng Sáu cho biết họ tìm hiểu về cáo buộc rằng ông Manning cung cấp video mật và 260.000 điện mật ngoại giao cho Wikileaks.

Ông Lewis cho biết Bộ Quốc phòng Mỹ, giống như bất kỳ tổ chức nào, đều có "những vai xấu" ở bên trong chống lại người tuyển dụng họ "nhưng nay để làm những việc như thế này thì đối với họ dễ dàng hơn rất nhiều."

Một cựu quan chức Lầu Năm Góc cho biết cuộc cách mạng truyền thông kỹ thuật số, trong khi mang lại lợi ích to lớn cho xã hội nói chung, cũng tạo ra lo ngại về an ninh. "Sự gia tăng của phương tiện truyền thông kỹ thuật số và phần mềm xã hội chắc chắn sẽ làm tăng rủi ro dẫn tới những vụ việc như thế này xảy ra", quan chức này nói với điều kiện không nêu tên vì ông vẫn còn đóng một vai trò tích cực trong mảng chính sách an ninh quốc gia.

Giới chuyên gia cho rằng trong thời đại dùng giấy thì một tài liệu có đóng dấu mật là đủ nhưng với thời kỹ thuật số thì mọi chuyện lại khác.

James Lewis, chuyên gia về không gian mạng, tại Trung tâm Chiến lược và Nghiên cứu Quốc tế (CSIS), đưa ra quan điểm trong thời đại Internet "nhiều người có thể truy cập cơ sở dữ liệu và xem tất cả tài liệu được lưu trữ một nơi nào đó" … "Nhưng cách chúng ta kiểm soát quyền truy cập lại dựa trên một mô hình cũ hơn, tức là dựa vào mức độ tin tưởng cá nhân"… "Lầu Năm Góc tin tưởng nhân viên của mình, đó là điều tốt, nhưng không đủ."

Lewis cho biết một "hệ thống tốt hơn có thể thông báo ngay việc tại sao có ai đó lại có thể tải xuống hàng ngàn tư liệu"?

Don Jackson từ SecureWorks cho biết trước khi có Internet thì người ta không quá lo lắng về việc dữ kiện bị phát tán bởi một tờ báo có đăng thì cũng không thể phát tán được 90.000 văn bản, thế nhưng Wikileaks có thể làm điều đó trong vài giây. Cựu quan chức của Lầu Năm Góc nói rằng ông "rất tiếc" đã xảy ra việc phát tán thông tin mật về Áp-ga-ni-xtan và Pa-ki-xtan, nhưng nói ông hy vọng vụ việc này sẽ không dẫn tới khả năng quân đội bớt sử dụng các phương tiện truyền thông xã hội. "Vụ việc này không nên được dùng để biện minh cho nỗ lực làm người ta bớt nắm bắt phương tiện truyền thông mới cũng như để khám phá ra cách sử dụng truyền thông mới một cách hiệu quả hơn", ông nói.

Mặt khác, việc tăng lên nhiều con số nhân viên dân sự và quân sự được tiếp xúc các nguồn tin mật cũng dễ gây thất thoát tư liệu. Hiện nay có hơn 800 nghìn người Mỹ có quyền xem các nguồn tin mật.

Báo New York Times gần đây trích các nghiên cứu của Hoa Kỳ cho rằng hiện nước này có hàng trăm cơ quan chính phủ cùng quản lý việc an ninh chống khủng bố.

2

MÃ HÓA KHÓA ĐỐI XỨNG

2.1. KHÁI NIỆM

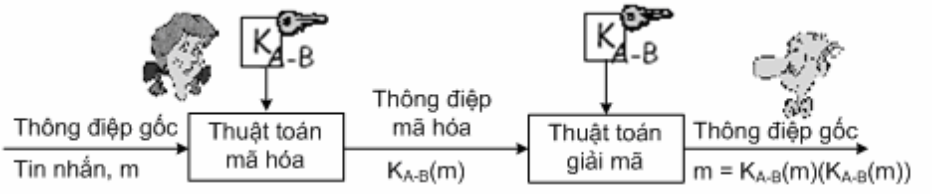
2.1.1. Mã hóa khóa đối xứng là gì?

Mã hóa khóa đối xứng (hay còn gọi là mã hóa khóa đồng bộ) là một thuật toán mà trong đó cả hai quá trình mã hóa và giải mã đều dùng một khóa. Để đảm bảo tính an toàn, khóa này phải được giữ bí mật. Vì thế các thuật toán mã hóa khóa đồng bộ này còn có tên gọi khác là mã hóa với khóa bí mật (*secret key cryptography*). Một điều cần lưu ý là khi một người mã hóa một thông điệp gốc (plaintext) thành thông điệp mã hóa bằng một khóa K (thuật toán mã hóa) (ciphertext) rồi gửi ciphertext cho đối tác thì đối tác muốn giải mã cũng cần phải có khóa K, nghĩa là trước đó hai đối tác đã phải trao đổi cho nhau chia sẻ để cùng biết được khóa K.

Trong ví dụ về gậy mã hóa của người Sparte, các đối tác phải bàn giao cho nhau để sở hữu những cây gậy giống nhau trước khi trao đổi thông điệp. Caesar muốn cho tướng lĩnh dưới quyền đọc được mật thư của mình thì trước khi ra đi các tướng lĩnh phải được Hoàng Đế triệu tập vào phòng kín để báo cho biết số bước xoay vòng và tất nhiên điều này (chìa khóa) phải được giữ kín!

Giả sử nếu An chỉ gửi thông điệp đã mã hóa cho Bình mà không hề báo trước về thuật toán mã hóa đã sử dụng, Bình sẽ chẳng hiểu trong thông điệp của An muốn nói gì. Vì thế bắt buộc An phải thông

báo cho Bình về chìa khóa và thuật toán sử dụng tại một thời điểm nào đó trước đấy.



Hình 2.1: Thuật toán mã hóa đối xứng

Bình và An có cùng một khóa K_{A-B}. Giả sử *m* là thông điệp gốc, khóa này được xây dựng sao cho: $m = K_{A-B}(K_{A-B}(m))$: dùng K_{A-B} vừa để mã hóa vừa để giải mã.

2.1.2. Mã hóa khóa đối xứng có thể phân thành hai nhóm phụ

- **Thuật toán mã hóa theo khối (*Block ciphers*)**: trong đó từng khối dữ liệu trong văn bản gốc ban đầu được thay thế bằng một khối dữ liệu khác có cùng độ dài. Độ dài mỗi khối gọi là kích thước khối (*block size*), thường được tính bằng đơn vị bit. Ví dụ thuật toán 3-Way có kích thước khối bằng 96 bit. Một số thuật toán khối thông dụng là: DES, 3DES, RC5, RC6, 3-Way, CAST, Camelia, Blowfish, MARS, Serpent, Twofish, GOST...

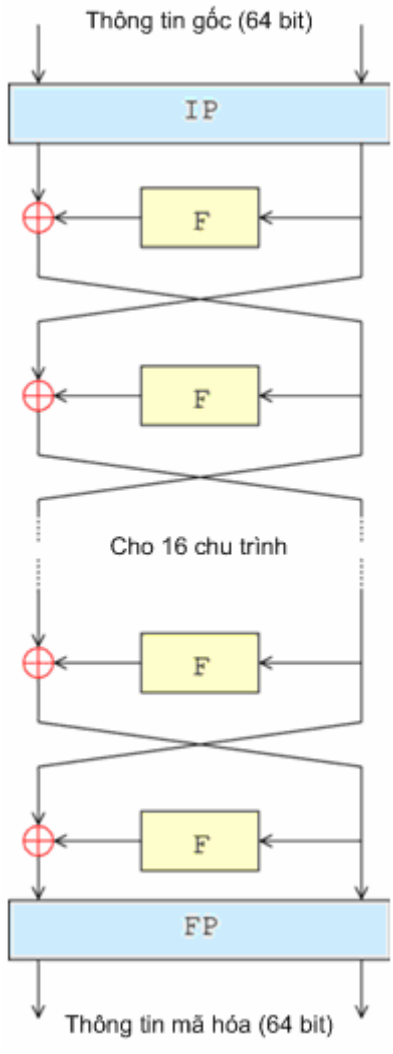
- **Thuật toán mã hóa dòng (*Stream ciphers*):** trong đó dữ liệu đầu vào được mã hóa từng bit một. Các thuật toán dòng có tốc độ nhanh hơn các thuật toán khối, được dùng khi khối lượng dữ liệu cần mã hóa chưa được biết trước, ví dụ trong kết nối không dây. Có thể coi thuật toán dòng là thuật toán khối với kích thước mỗi khối là 1 bit. Một số thuật toán dòng thông dụng: RC4, A5/1, A5/2, Chameleon.

2.2. TIÊU CHUẨN MÃ HÓA DỮ LIỆU (DES)

2.2.1. Giới thiệu về DES

Tiêu chuẩn mã hóa dữ liệu DES (Data Encryption Standard) là một phương pháp mật mã hóa được FIPS (Federal Information

Processing Standard: Tiêu chuẩn xử lý thông tin Liên bang Hoa Kỳ) chọn làm chuẩn chính thức vào năm 1976. Thuật toán mã hóa theo tiêu chuẩn DES gọi là DEA (Data Encryption Algorithm). (Người ta cũng thường gọi lẫn lộn DEA và DES trong khi sử dụng). DES là một mã khối, mỗi khối gồm 64 bit trong đó dành 8 bit để kiểm tra lỗi (Parity checking) còn lại 56 bit khóa (xem Phụ lục 1). Cấu trúc tổng thể của thuật toán được thể hiện ở hình 2.2.



Hình 2.2: Mô hình thuật toán DES

Mô tả thuật toán DES

Có 16 chu trình giống nhau trong quá trình xử lý. Ngoài ra còn có hai lần hoán vị đầu và cuối (*Initial and final permutation:* IP & FP). Hai quá trình này có tính chất đối nhau (trong quá trình mã hóa thì IP trước FP, khi giải mã thì ngược lại). IP và FP, được sử dụng từ thập niên 1970, không có vai trò xét về mật mã học và việc sử dụng chúng chỉ có ý nghĩa đáp ứng cho quá trình đưa thông tin vào và lấy thông tin ra từ các khối phần cứng.

Trước khi đi vào 16 chu trình chính, khối thông tin 64 bit được tách làm hai phần 32 bit và mỗi phần sẽ được xử lý tuần tự (quá trình này còn được gọi là mạng Feistel). Cấu trúc của thuật toán (mạng Feistel) đảm bảo rằng quá trình mã hóa và giải mã diễn ra tương tự. Điểm khác nhau chỉ ở chỗ các khóa con được sử dụng theo trình tự ngược nhau. Điều này giúp cho việc thực hiện thuật toán trở nên đơn giản, đặc biệt là khi thực hiện bằng phần cứng.

Ký hiệu ⊕ (trong hình 2.2) thể hiện phép toán XOR (hàm "tuyển ngặt": *Exclusive OR*) hay là hàm "cộng theo modulo 2". Hàm F làm biến đổi một khối 32 bit đang xử lý với một khóa con.

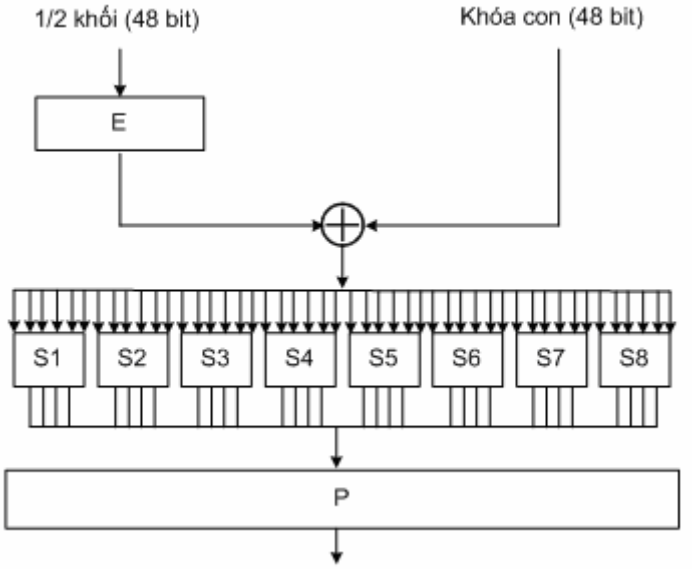
Đầu ra sau hàm F được kết hợp khối 32 bit còn lại và hai phần được tráo đổi để xử lý trong chu trình kế tiếp. Sau chu trình cuối cùng thì 2 nửa không bị tráo đổi; đây là đặc điểm của cấu trúc Feistel khiến cho quá trình mã hóa và giải mã trở nên giống nhau.

Hàm Feistel (F)

Hàm F, như được miêu tả như hình 2.3, hoạt động trên khối 32 bit và bao gồm bốn giai đoạn:

1. *Mở rộng*: 32 bit đầu vào được mở rộng thành 48 bit sử dụng thuật toán hoán vị mở rộng (*expansion permutation*) với việc nhân đôi một số bit. Giai đoạn này được ký hiệu là E trong sơ đồ.

2. *Trộn khóa*: 48 bit thu được sau quá trình mở rộng được XOR với khóa con. Mười sáu khóa con 48 bit được tạo ra từ khóa chính 56 bit theo một chu trình tạo khóa con (*key schedule*) miêu tả ở phần sau. (Xem khái niệm hàm XOR ở phụ lục I)
3. *Thay thế*: 48 bit sau khi trộn được chia làm 8 khối con 6 bit và được xử lý qua hộp thay thế S-box. Đầu ra của mỗi khối 6 bit là một khối 4 bit theo một chuyển đổi phi tuyến được thực hiện bằng một bảng tra. Khối S-box đảm bảo phần quan trọng cho độ an toàn của DES. Nếu không có S-box thì quá trình sẽ là tuyến tính và việc thám mã sẽ rất đơn giản.
4. *Hoán vị*: Cuối cùng, 32 bit thu được sau S-box sẽ được sắp xếp lại theo một thứ tự cho trước (còn gọi là P-box).

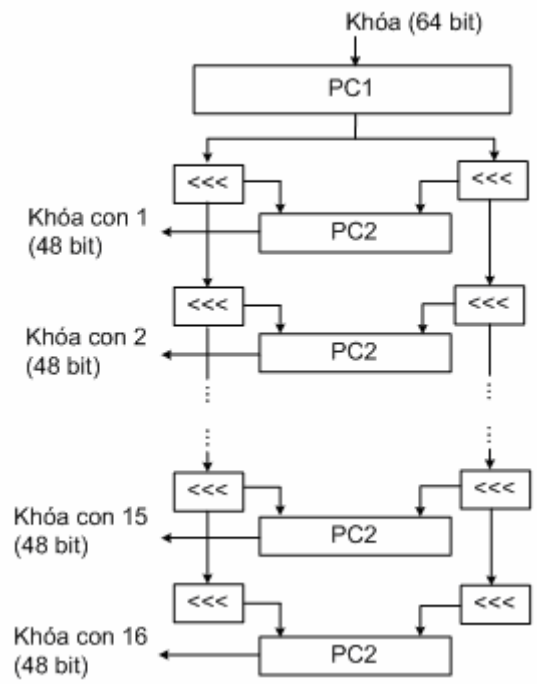


Hình 2.3: Hàm F (Feistel) dùng trong DES

Quá trình luân phiên sử dụng S-box và sự hoán vị các bit cũng như quá trình mở rộng đã thực hiện được tính chất gọi là sự xáo

trộn và khuếch tán (*confusion and diffusion*). Đây là yêu cầu cần có của một thuật toán mã hóa được Claude Shannon phát hiện trong những năm 1940.

Quá trình tạo khóa con (Sub-key)



Hình 2.4: Quá trình tạo khóa con trong DES

Hình 2.4 mô tả thuật toán tạo khóa con cho các chu trình. Đầu tiên, từ 64 bit ban đầu của khóa, 56 bit được chọn (*Permuted Choice 1*, hay PC-1); 8 bit còn lại bị loại bỏ. 56 bit thu được được chia làm hai phần bằng nhau, mỗi phần được xử lý độc lập. Sau mỗi chu trình, mỗi phần được dịch đi 1 hoặc 2 bit (tùy thuộc từng chu trình).

Các khóa con 48 bit được tạo thành bởi thuật toán lựa chọn 2 (*Permuted Choice 2*, hay PC-2) gồm 24 bit từ mỗi phần. Quá trình dịch chuyển bit (được ký hiệu là "<<<" trong sơ đồ) khiến cho các khóa con sử dụng các bit khác nhau của khóa chính; mỗi bit được sử dụng trung bình là 14 trong tổng số 16 khóa con.

Quá trình tạo khóa con khi thực hiện giải mã cũng diễn ra tương tự nhưng các khóa con được tạo theo thứ tự ngược lại. Ngoài ra sau mỗi chu trình, khóa sẽ được dịch chuyển phải thay vì dịch chuyển trái như khi mã hóa.

2.2.2. Sự ra đời của DES

Cho đến trước những năm 60 của thế kỷ XX, công nghệ bảo mật thông tin hầu như là độc quyền của các cơ quan an ninh quốc phòng của các Nhà nước, chẳng hạn như ở Mỹ là Cơ quan an ninh quốc gia NSA (National Security Agency). Từ thập kỷ 70 của thế kỷ XX, nhu cầu giao dịch xã hội và kinh tế trên phạm vi toàn cầu đòi hỏi một sự phát triển mạnh mẽ về lĩnh vực bảo mật thông tin, cụ thể là trong các vấn đề lập mã và giải mã. Nhiều công ty ra đời và phát triển nhiều công cụ bảo mật nhưng không có một sự thẩm định đáng tin cậy nào cho những công cụ đó.

Cuối cùng đến năm 1972, Viện quốc gia về tiêu chuẩn và công nghệ, nay là Viện quốc gia về tiêu chuẩn NIST (National Institute of Standards and Technolgy) của Mỹ quyết định chủ trì vấn đề này và đề xuất việc xây dựng một tiêu chuẩn quốc gia về bảo mật dữ liệu lấy tên là Tiêu chuẩn mã hóa dữ liệu (quốc gia) DES (Data Encryption Standard) và năm 1974 NIST đã chọn một thuật toán mã hóa do IBM giới thiệu làm thuật toán đạt tiêu chuẩn và gắn tên cho thuật toán đó là Thuật toán mã hóa tiêu chuẩn DEA (Data Encryption Algorithm).

Ý tưởng chính của thuật toán DEA do một nhà lập trình của IBM là Horst Feistel sáng tạo, là việc thực hiện lặp nhiều chu trình mã hóa bằng cả các luật thay thế và các luật chuyển vị của mã hóa cổ điển. Trước kia, chỉ với các công cụ cơ giới việc thực hiện lặp các quá trình chuyển vị rất khó khăn nên các công cụ mã hóa phức tạp trước đây (như máy Enigma) chỉ sử dụng các thay thế lặp, không dùng chuyển vị. Sau năm 1970 với sự phát triển của máy tính điện tử, Feistel đã thực hiện được điều đó cho nên độ phức tạp của DEA trội hẳn so với các thuật toán mã hóa trước đây. NIST đã yêu cầu NSA trợ giúp phát

triển DEA và NSA đã đáp ứng. Tuy nhiên có người cho rằng NSA đã đề nghị giảm độ dài khóa do IBM đưa ra lúc ban đầu là 128 bit xuống chỉ còn 56 bit sau này là vì lo ngại mức độ bảo mật quá cao, vượt khỏi trình độ khống chế của NSA thời đó và như thế có khả năng ảnh hưởng đến vấn đề an toàn bảo mật của quốc gia.

NSA cũng đề nghị chỉ sản xuất các phần cứng tích hợp phần mềm bảo mật DEA để phổ biến trên thị trường nhưng không được phổ biến các kết quả nghiên cứu về phần mềm. Tuy nhiên, dù có sự phản ứng (không công khai) của NSA, kết quả là DEA vẫn được công nhận là một phần mềm mã hóa đạt tiêu chuẩn mã hóa dữ liệu quốc gia của Mỹ dành cho việc bảo mật các thông tin dữ liệu kinh tế và xã hội, không thuộc phạm vi được quy định là TUYỆT MẬT của Nhà nước. Từ đó DEA nhanh chóng phát triển và phổ biến rộng khắp, không những chỉ ở Mỹ mà còn lan rộng khắp toàn thế giới. Có thể nói rằng từ xưa đến nay chưa có một thuật toán mã hóa nào được thừa nhận và sử dụng phổ biến rộng rãi trên thế giới trong một thời gian dài như vậy.

Từ năm 1977 NIST phổ biến công khai tiêu chuẩn DES và quy định cứ sau 5 năm sẽ xem xét lại một lần. Vào các năm 1983, 1987 và 1993 DES đều được công nhận gia thời hạn sử dụng thêm 5 năm tiếp sau.

Cho đến năm 1997, do sự phát triển tốc độ của máy tính điện tử và những kết quả nghiên cứu mới về thám mã, DES bắt đầu bộc lộ những bất cập và NIST đặt vấn đề tìm cách thay thế DES bằng các thuật toán mã hóa mới có độ bảo mật cao hơn qua các kỳ thi tuyển chọn các thuật toán mã hóa tiên tiến AEA (Advanced Encryption Algorithm).

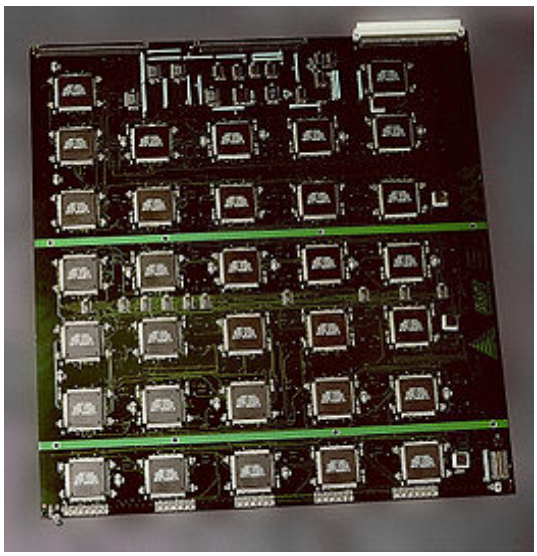
2.2.3. An toàn và sự giải mã

Thuật toán DES được sử dụng là một chuẩn mã hóa trong thương mại và mặc dù đã có nhiều nghiên cứu về phá mã DES hơn bất kỳ phương pháp mã hóa khối nào khác, nhưng phương pháp phá

mã thực tế nhất hiện nay vẫn là tấn công bằng bạo lực. Nhiều đặc tính mật mã hóa của DES đã được xác định và từ đó đã có ba phương pháp phá mã khác được xác định với mức độ phức tạp nhỏ hơn tấn công bạo lực, tuy nhiên các phương pháp này đòi hỏi một số lượng plaintext quá lớn (để tấn công lựa chọn tần suất trong plaintext) nên hầu như không thực hiện được trong thực tế.

Tấn công bạo lực (Bruce force attack)

- Đối với bất cứ phương pháp mã hóa nào, kiểu tấn công cơ bản nhất là tấn công bằng bạo lực: thử lần lượt tất cả các khóa có thể cho đến khi tìm ra khóa đúng. Độ dài của khóa sẽ xác định số lượng phép thử tối đa cần thực hiện và do đó thể hiện tính khả thi của phương pháp. Trong trường hợp của DES, nghi ngờ về độ an toàn của nó đã được đặt ra ngay từ khi nó chưa trở thành tiêu chuẩn. Người ta cho rằng chính NSA đã ủng hộ IBM giảm độ dài khóa từ 128 bit xuống 64 bit và tiếp tục xuống 56 bit. (*Điều này dẫn đến suy đoán rằng NSA đã có thể có hệ thống tính toán đủ mạnh để phá vỡ khóa 56 bit ngay từ những năm 1970*).



Hình 2.5. Mô tả sơ đồ phá mã

Hệ thống phá mã DES của Hiệp hội EFF được xây dựng với ngân sách 250.000 USD (vào thời đó). Hệ thống bao gồm 1536 bộ vi xử lý thiết kế riêng và có khả năng duyệt hết mọi khóa DES trong vòng vài ngày. Hình 2.5 thể hiện một phần bảng mạch của hệ thống chứa một vài bộ vi xử lý.

Trong giới nghiên cứu, nhiều đề xuất về các hệ thống phá mã DES được đề ra. Năm 1977 Diffie và Hellman dự thảo một hệ thống có giá khoảng 20 triệu USD và có khả năng phá khóa DES trong 1 ngày. Năm 1993, Wiener dự thảo một hệ thống khác có khả năng phá mã trong vòng 7 giờ với giá 1 triệu USD.

Những điểm yếu của DES được thực sự chứng minh vào cuối những năm 1990. Vào năm 1997, công ty bảo mật RSA đã tài trợ một loạt những cuộc thi với giải thưởng 10.000 USD cho đội đầu tiên phá mã được một bản tin mã hóa bằng DES. Đội chiến thắng trong cuộc thi này là dự án DESCHALL với những người lãnh đạo là Rocke Verser, Matt Curtin và Justin Dolske. Họ đã sử dụng hàng nghìn máy tính nối mạng để phá mã.

Khả năng phá mã DES được chứng minh thêm lần nữa vào năm 1998 khi tổ chức *Electronic Frontier Foundation* (EFF), một tổ chức hoạt động cho quyền công dân trên Internet, xây dựng một hệ thống chuyên biệt để phá mã với giá thành 250.000 USD. Động cơ thúc đẩy EFF trong hành động này là nhằm chứng minh DES có thể bị phá vỡ trên lý thuyết cũng như trên thực tế: "Nhiều người không tin vào chân lý này cho đến khi họ nhìn thấy sự việc bằng chính mắt mình. Xây dựng một bộ máy có thể phá khóa DES trong vòng vài ngày là cách duy nhất chứng tỏ với mọi người rằng họ không thể đảm bảo an ninh thông tin dựa vào DES."

Hệ thống này đã tìm được khóa DES bằng phương pháp bạo lực trong thời gian hơn 2 ngày; trong khi vào khoảng thời gian đó, một nhà lãnh đạo của Bộ Tư pháp Hoa Kỳ (DOJ) vẫn tuyên bố rằng DES là không thể bị phá vỡ.

Các kiểu tấn công khác hiệu quả hơn phương pháp bạo lực

Hiện nay có 3 kiểu tấn công có khả năng phá vỡ DES (với đủ 16 chu trình) với độ phức tạp khá thấp: phá mã vi sai DC (*Differential Cryptanalysis*), phá mã tuyến tính LC (*Linear Cryptanalysis*) và phá mã Davies (*Davies' attack*). Tuy nhiên các dạng tấn công này chưa thực hiện được trong thực tế.

- **Phá mã vi sai**, đòi hỏi dùng 2^{47} *plaintexts* được xem là do Eli Biham và Adi Shamir tìm ra vào cuối những năm 1980 mặc dù đã được IBM và NSA biết đến trước đó để phá mã DES với đủ 16 chu trình (nhưng chưa có công bố chính thức).

- Phá mã tuyến tính được tìm ra bởi Mitsuru Matsui, đòi hỏi 2^{43} *plaintexts* (Matsui, 1993). Phương pháp này đã được Matsui thực hiện và là cuộc thực nghiệm phá mã đầu tiên được công bố. Không có bằng chứng chứng tỏ DES có khả năng chống lại tấn công dạng này.

Một phương pháp tổng quát hơn, **phá mã tuyến tính đa chiều** (*multiple linear cryptanalysis*), được Kaliski và Robshaw nêu ra vào năm 1994, sau đó Biryukov và cộng sự tiếp tục cải tiến vào năm 2004. Nghiên cứu của họ cho thấy mô phỏng tuyến tính đa chiều có thể sử dụng để giảm độ phức tạp của quá trình phá mã tới 4 lần (chỉ còn 2^{41} *plaintexts*).

- **Phá mã Davies**: trong khi phá mã vi sai và phá mã tuyến tính là các kỹ thuật phá mã tổng quát, có thể áp dụng cho các thuật toán khác nhau, phá mã Davies là một kỹ thuật dành riêng cho DES. Dạng tấn công này được đề xuất lần đầu bởi Davies vào cuối những năm 1980 và cải tiến bởi Biham và Biryukov (1997). Dạng tấn công mạnh đòi hỏi 2^{50} *plaintexts,* độ phức tạp là 2^{50} và tỷ lệ thành công là 51%.

Ngoài ra còn có những kiểu tấn công dựa trên bản thu gọn của DES (DES với ít hơn 16 chu trình). Những nghiên cứu này cho chúng ta biết số lượng chu trình cần có và ranh giới an toàn của hệ thống. Năm 1994, Langford và Hellman đề xuất phá mã vi sai tuyến tính

(*differential-linear cryptanalysis*) kết hợp giữa phá mã vi sai và tuyến tính. Một dạng cải tiến của phương pháp này có thể phá vỡ DES 9 chu trình với $2^{15,8}$ plaintexts và có độ phức tạp là $2^{29,2}$ (Biham et al, 2002).

Tháng 6/1997 dự án DESCHALL đã phá vỡ được một bản tin mã hóa bằng DES lần đầu tiên trước công chúng. Thiết bị thám mã DEEP CRACK của tổ chức Electronic Foundation phá được một khóa của DES trong vòng 56 giờ và đến tháng 01/1999 đã cùng với *distributed.net* phá được một khóa chỉ trong vòng 22 giờ 15 phút.

* Để tăng độ an toàn, người sử dụng DES trước đây chuyển sang dùng Double DES và Triple DES (2DES và TDES). 2DES thực hiện 2 lần thuật toán mã hóa DEA với hai khóa riêng biệt, tăng độ dài khóa từ 56 lên 112 bit. Thoạt đầu người ta nghĩ rằng, theo tính toán thì tăng thêm 1 bit của độ dài khóa thì độ phức tạp của khóa (số trường hợp phải duyệt trong tấn công bạo lực) tăng gấp đôi. Và như vậy thì độ phức tạp khóa trong 2DES lên đến 2^{56} lần so với khóa trong DES! Nhưng Whitfield Diffie và Martin Hellman đã phát minh ra một phương pháp thám mã gọi là tấn công gặp tại điểm giữa (*meet–in–the–middle attack*) làm cho độ phức tạp của 2DES chỉ tăng gấp đôi của DES tức là chỉ bằng: $2.2^{56} = 2^{57}$. Triple DES cũng sử dụng DES ba lần cho một plaintext với những khóa khác nhau để làm tăng độ dài khóa lên. Hiện nay Triple DES được xem là đủ an toàn mặc dù tốc độ thực hiện quá chậm.

2.2.3. Một vài đặc điểm về cách giải mã

Thuật toán mã hóa theo chuẩn DES có tính chất bù nghĩa là:

$$E_K(P) = C \Leftrightarrow E_{\overline{K}}(\overline{P}) = \overline{C}$$

trong đó $\overline{x}$ là phần bù của x theo từng bit (1 thay bằng 0 và ngược lại). $\boldsymbol{E_K}$ là bản mã hóa của E với khóa K. $\boldsymbol{P}$ và $\boldsymbol{C}$ là plaintext (trước khi mã hóa) và ciphertext (sau khi mã hóa). Do tính bù, ta có thể giảm độ phức tạp của tấn công bạo lực xuống 2 lần (tương ứng với 1 bit) với điều kiện là ta có thể lựa chọn plaintext.

Ngoài ra DES còn có **4 khóa yếu** (*weak keys*). Khi sử dụng khóa yếu thì mã hóa (E) và giải mã (D) sẽ cho ra cùng kết quả:

$$E_K(E_K(P)) = P \text{ hoặc tương đương } E_K = D_K$$

Bên cạnh đó, còn có 6 cặp **khóa nửa yếu** *(semi-weak keys*). Mã hóa với một khóa trong cặp K_1, tương đương với giải mã với khóa còn lại K_2:

$$E_{K1}(E_{K2}(P) = P \text{ hoặc tương đương } E_K = D_K$$

Tuy nhiên có thể dễ dàng tránh được những khóa này khi thực hiện thuật toán, có thể bằng cách thử hoặc chọn khóa ngẫu nhiên thì khả năng chọn phải khóa yếu là rất nhỏ.

DES đã được chứng minh là không tạo thành nhóm. Nói một cách khác, tập hợp $\{E_K\}$ (cho tất cả các khóa có thể) với phép hợp thành U không tạo thành một nhóm hay nhiều nhóm *(pseudo-group)* (kết quả của Campbell and Wiener, 1992).

Vấn đề này đã từng là một câu hỏi mở trong khá lâu. Nếu như tạo thành nhóm thì DES có thể bị phá vỡ dễ dàng hơn bởi vì việc áp dụng DES nhiều lần (ví dụ như trong 2DES, Triple DES) sẽ không làm tăng thêm độ an toàn của DES.

2.3. TIÊU CHUẨN MÃ HÓA TIÊN TIẾN (AES)

2.3.1. Sự ra đời của AES

Từ cuối thập niên 1980, đầu thập niên 1990, xuất phát từ những lo ngại về độ an toàn và tốc độ thấp khi áp dụng bằng phần mềm, giới nghiên cứu đã đề xuất khá nhiều thuật toán mã hóa khối để thay thế DES. Những ví dụ tiêu biểu bao gồm: RC5, Blowwfish, IDEA (International Data Encryption Algorithm: Thuật toán mã hóa dữ liệu quốc tế), NewDES, SAFER và FEAL. Hầu hết những thuật toán này có thể sử dụng từ khóa 64 bit của DES mặc dù chúng thường được thiết kế hoạt động với từ khóa 64 bit hay 128 bit. Bản thân DES cũng cải tiến để có thể được sử dụng an toàn hơn.

Năm 2001, sau một cuộc thi quốc tế, NIST đã chọn ra một thuật toán mới là Tiêu chuẩn mã hóa tiên tiến AES (*Advanced Encryption Standard*) để thay thế cho DES. Thuật toán được trình diện dưới tên là *Rijndael*. Những thuật toán khác có tên trong danh sách cuối cùng của cuộc thi AES gồm: *RC6, Serpent, MARS và Twofish*. *AES* là thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa thay cho tiêu chuẩn DES trước đó. Giống như tiêu chuẩn DES, AES được kỳ vọng áp dụng trên phạm vi toàn thế giới và đã được nghiên cứu rất kỹ lưỡng. AES được chấp thuận làm tiêu chuẩn liên bang bởi Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST) trong một quá trình tiêu chuẩn hóa kéo dài 5 năm.

Thuật toán được thiết kế bởi hai nhà mật mã học người Bỉ: Joan Daemen và Vincent Rijmen. Thuật toán được đặt tên là "**Rijndael**" khi tham gia cuộc thi thiết kế AES theo cách ghép tên của hai đồng tác giả. Thuật toán được dựa trên bản thiết kế Square có trước đó của Daemen và Rijmen; còn Square lại được thiết kế dựa trên Shark. Khác với DES sử dụng mạng Feistel, Rijndael sử dụng mạng thay thế-chuyển vị. AES có thể dễ dàng thực hiện với tốc độ cao bằng phần mềm hoặc phần cứng và không đòi hỏi nhiều bộ nhớ. Do là một tiêu chuẩn mã hóa mới, AES đang được triển khai sử dụng rộng rãi hàng loạt.

2.3.2. Mô tả thuật toán

Mặc dù 2 tên *AES* và *Rijndael* vẫn thường được gọi thay thế cho nhau nhưng trên thực tế thì 2 thuật toán không hoàn toàn giống nhau. AES chỉ làm việc với khối dữ liệu 128 bit và khóa có độ dài 128, 192 hoặc 256 bit trong khi Rijndael có thể làm việc với dữ liệu và khóa có độ dài bất kỳ là bội số của 32 bit nằm trong khoảng từ 128 tới 256 bit. Các khóa con sử dụng trong các chu trình được tạo bởi quá trình tạo khóa con Rijndael. Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn. AES làm

việc với từng khối dữ liệu 4×4 bytes (tiếng Anh: *state*, khối trong Rijndael có thể có thêm cột). Quá trình mã hóa gồm 4 bước:

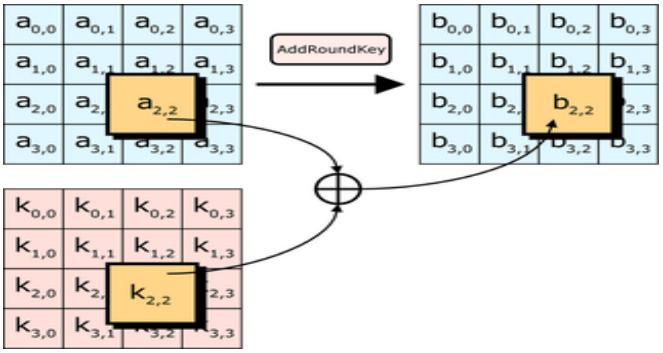
1. AddRoundKey: mỗi byte của khối được kết hợp với khóa con, các khóa con này được tạo ra từ quá trình tạo khóa con Rijndael.
2. SubBytes: đây là phép thế (phi tuyến) trong đó mỗi byte sẽ được thế bằng một byte khác theo bảng tra (Rijndael S-box).
3. ShiftRows: đổi chỗ, các hàng trong khối được dịch vòng.
4. MixColumns: quá trình trộn làm việc theo các cột trong khối theo một phép biến đổi tuyến tính. Tại chu trình cuối thì bước *MixColumns* được thay thế bằng bước *AddRoundKey*.

Bước AddRoundKey. Tại bước này, khóa con được kết hợp với các khối. Khóa con trong mỗi chu trình được tạo ra từ khóa chính với quá trình tạo khóa con Rijndael; mỗi khóa con có độ dài giống như các khối. Quá trình kết hợp được thực hiện bằng cách XOR từng bit của khóa con với khối dữ liệu.

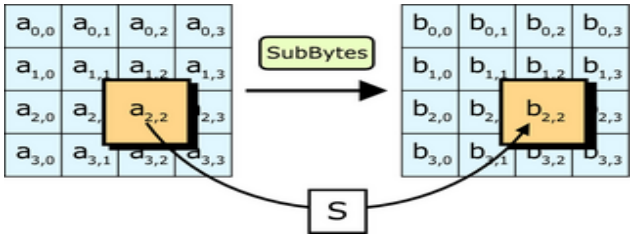
Bước SubBytes. Các byte được thế thông qua bảng tra S-box. Đây chính là quá trình phi tuyến của thuật toán. Hộp S-box này được tạo ra từ một phép nghịch đảo trong trường hữu hạn GF (2^8) có tính chất phi tuyến. Để chống lại các tấn công dựa trên các đặc tính đại số, hộp S-box này được tạo nên bằng cách kết hợp phép nghịch đảo với một phép biến đổi affine khả nghịch. Hộp S-box này cũng được chọn để tránh các điểm bất động (fixed point).

Bước ShiftRows. Các hàng được dịch vòng một số vị trí nhất định. Đối với AES, hàng đầu được giữ nguyên. Mỗi byte của hàng thứ 2 được dịch trái một vị trí. Tương tự, các hàng thứ 3 và 4 được dịch 2 và 3 vị trí. Do vậy, mỗi cột khối đầu ra của bước này sẽ bao gồm các byte ở đủ 4 cột khối đầu vào. Đối với Rijndael với độ dài khối khác nhau thì số vị trí dịch chuyển cũng khác nhau.

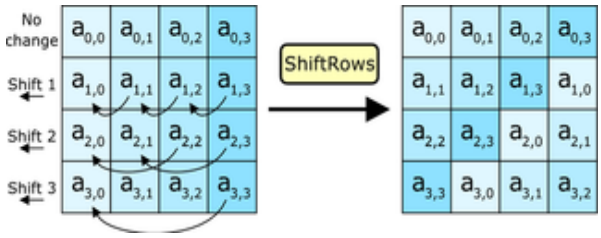
Bước MixColumns. Bốn byte trong từng cột được kết hợp lại theo một phép biến đổi tuyến tính khả nghịch. Mỗi khối 4 byte đầu vào sẽ cho một khối 4 byte ở đầu ra với tính chất là mỗi byte ở đầu vào đều ảnh hưởng tới cả 4 byte đầu ra. Cùng với bước ShiftRows, MixColumns đã tạo ra tính chất khuếch tán cho thuật toán. Mỗi cột được xem như một đa thức trong trường hữu hạn và được nhân với đa thức $c(x) = 3x^3 + x^2 + x + 2$ (modulo $x^4 + 1$). Vì thế, bước này có thể được xem là phép nhân ma trận trong trường hữu hạn.



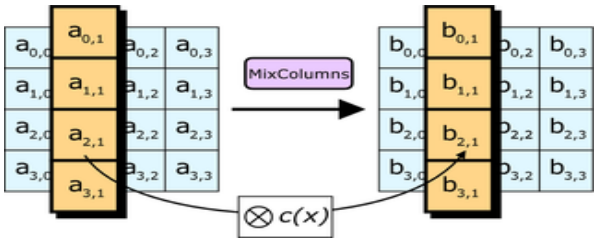
Trong bước **AddRoundKey,** mỗi byte được kết hợp với một byte trong khóa con của chu trình sử dụng phép toán XOR.



Trong bước **SubBytes,** mỗi byte được thay thế bằng một byte theo bảng tra, S; $b_{ij} = S(a_{ij})$.



Trong bước **ShiftRows**, các byte trong mỗi hàng dịch vòng trái. Số vị trí dịch chuyển tùy từng hàng.



Trong bước **MixColumns**, mỗi cột được nhân với một hệ số cố định c(x).

Hình 2.6: Sơ đồ thuật toán AES

2.3.3. Tối ưu hóa

Đối với các hệ thống 32 bit hoặc lớn hơn, ta có thể tăng tốc độ thực hiện thuật toán bằng cách sát nhập các bước *SubBytes, ShiftRows, MixColumns* và chuyển chúng thành dạng bảng. Có cả thảy 4 bảng với 256 mục, mỗi mục là 1 từ 32 bit, 4 bảng này chiếm 4096 byte trong bộ nhớ. Khi đó, mỗi chu trình sẽ được bao gồm 16 lần tra bảng và 12 lần thực hiện phép XOR 32 bit cùng với 4 phép XOR trong bước *AddRoundKey*.

Trong trường hợp kích thước các bảng vẫn lớn so với thiết bị thực hiện thì chỉ dùng một bảng và tra bảng kết hợp với hoán vị vòng quanh.

2.3.4. Độ an toàn của AES

Vào thời điểm năm 2006, dạng tấn công lên AES duy nhất thành công là tấn công kênh bên (*side channel attack*). Vào tháng 6 năm 2003, Chính phủ Hoa Kỳ tuyên bố AES có thể được sử dụng cho thông tin mật.

> *"Thiết kế và độ dài khóa của thuật toán AES (128, 192 và 256 bit) là đủ an toàn để bảo vệ các thông tin được xếp vào loại TỐI MẬT (secret). Các thông tin TUYỆT MẬT (top secret) sẽ phải dùng khóa 192 hoặc 256 bit. Các phiên bản thực hiện AES nhằm mục đích bảo vệ hệ thống an ninh hay thông tin quốc gia phải được NSA kiểm tra và chứng nhận trước khi sử dụng."*

Điều này đánh dấu lần đầu tiên công chúng có quyền tiếp xúc với thuật toán mật mã mà NSA phê chuẩn cho thông tin TUYỆT MẬT.

Nhiều phần mềm thương mại hiện nay sử dụng mặc định khóa có độ dài 128 bit.

Phương pháp thường dùng nhất để tấn công các dạng mã hóa khối là thử các kiểu tấn công lên phiên bản có số chu trình thu gọn. Đối với khóa 128 bit, 192 bit và 256 bit, AES có tương ứng 10, 12 và 14 chu trình. Tại thời điểm năm 2006, những tấn công thành công được biết đến là 7 chu trình đối với khóa 128 bit, 8 chu trình với khóa 192 bit và 9 chu trình với khóa 256 bit.

Một số nhà khoa học trong lĩnh vực mật mã lo ngại về an ninh của AES. Họ cho rằng ranh giới giữa số chu trình của thuật toán và số chu trình bị phá vỡ quá nhỏ. Nếu các kỹ thuật tấn công được cải thiện thì AES có thể bị phá vỡ. Ở đây, *phá vỡ* có nghĩa chỉ bất cứ phương pháp tấn công nào nhanh hơn tấn công kiểu duyệt toàn bộ (tấn công bạo lực).

Vì thế một tấn công cần thực hiện 2^{120} plaintexts cũng được coi là thành công mặc dù tấn công này chưa thể thực hiện trong thực tế. Tại thời điểm hiện nay, nguy cơ này không thực sự nguy hiểm và có thể bỏ qua.

Tấn công kiểu duyệt toàn bộ quy mô nhất đã từng thực hiện là do *distributed.net* thực hiện lên hệ thống 64 bit RC5 vào năm 2002 (Theo định luật Moore thì nó tương đương với việc tấn công vào hệ thống 66 bit hiện nay).

Một vấn đề khác nữa là cấu trúc toán học của AES. Không giống với các thuật toán mã hóa khác, AES có mô tả toán học khá đơn giản. Tuy điều này chưa dẫn đến mối nguy hiểm nào nhưng một số nhà nghiên cứu sợ rằng sẽ có người lợi dụng được cấu trúc này trong tương lai.

Vào năm 2002, Nicolas Courtois và Josef Pieprzyk phát hiện một tấn công trên lý thuyết gọi là **tấn công XSL** và chỉ ra điểm yếu tiềm tàng của AES. Tuy nhiên, một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề chưa rõ ràng trong cơ sở toán học của tấn công này và cho rằng các tác giả đã có thể có sai lầm trong tính toán.

Việc tấn công dạng này có thực sự trở thành hiện thực hay không vẫn còn để ngỏ và cho tới nay thì tấn công XSL vẫn chỉ là suy đoán.

2.3.5. Tấn công kênh bên (Side channel attacks)

Tấn công kênh bên không tấn công trực tiếp vào thuật toán mã hóa mà thay vào đó, tấn công lên các hệ thống thực hiện thuật toán có sơ hở làm lộ dữ liệu.

Tháng 4 năm 2005, Daniel J. Bernstein công bố một tấn công lên hệ thống mã hóa AES trong OpenSSL. Một máy chủ được thiết kế để đưa ra tối đa thông tin về thời gian có thể thu được và cuộc tấn công cần tới 200 triệu plaintexts lựa chọn. Một số người cho rằng tấn công không thể thực hiện được trên Internet với khoảng cách vài điểm mạng.

Tháng 10 năm 2005, Adi Shamir và 2 nhà nghiên cứu khác có một bài nghiên cứu minh họa một vài dạng khác. Trong đó, một tấn công có thể lấy được khóa AES với 800 lần ghi trong 65 mili giây.

Tấn công này yêu cầu kẻ tấn công có khả năng chạy chương trình trên chính hệ thống thực hiện mã hóa.

2.4. ƯU/NHƯỢC ĐIỂM VÀ PHẠM VI SỬ DỤNG CỦA MÃ HÓA ĐỐI XỨNG

Ưu điểm nổi bật của mã hóa đối xứng là tốc độ lập mã, giải mã khá nhanh chóng. Hiện nay có nhiều phần mềm thương mại hỗ trợ thuật toán mã hóa đối xứng hữu hiệu và rất phổ dụng.

Ưu điểm thứ hai là tuy có nhiều nghiên cứu thám mã đã thực hiện nhưng với các thuật toán được cải tiến gần đây như 3-DES và

nhất là AES thì độ bảo mật khá cao, trong thực tế việc phá mã cũng không dễ dàng.

Tuy vậy nhược điểm lớn nhất của thuật toán mã hóa đối xứng là vấn đề chuyển giao chìa khóa giữa các đối tác, đặc biệt là trong môi trường mở.

Như trong ví dụ nói ở đầu chương về việc trao đổi thông điệp giữa An và Bình. Bình nhận được thông điệp đã mã hóa của An, muốn giải mã được thì Bình phải có chìa khóa mã của An. An không thể chuyển giao khóa mã đồng thời với thông điệp vì như vậy thì việc mã hóa trở thành vô tác dụng.

Vì vậy An phải dùng một phương pháp nào đó để chuyển giao khóa giải mã cho Bình trước khi gửi thông điệp. Mà dù dùng phương thức thông tin nào trong môi trường mở: gửi thư, E-mail, gọi điện thoại v.v. thì vẫn có nguy cơ có người thứ ba nắm bắt được khóa mã và kết quả vẫn như thế!

So sánh lại với 5 nguyên lý bảo mật thông tin, xét trường hợp giao dịch của 2 đối tác An và Bình. Giả sử An và Bình "hoàn toàn tin tưởng vào nhau" và trao cho nhau mã khóa đối xứng bằng một phương pháp đáng tin cậy nào đó (trao tay trực tiếp hoặc có một phương pháp nào có thể thay thế cho trao tay trực tiếp mà cũng có giá trị tương đương như thế) và sau đó hai người sử dụng mã khóa truyền các thông điệp mã hóa cho nhau, ta thấy rằng:

- Sử dụng mã đối xứng (trong các điều kiện nói trên) đảm bảo được nguyên lý bí mật/riêng tư vì thông tin không thể bị lộ.

- Đảm bảo tính xác thực, tính không chối bỏ và tính nhận dạng, những điều này chủ yếu được thực hiện khi chuyển giao khóa mã cho nhau chứ không phải trong quá trình trao đổi thông điệp mã hóa về sau. Vì giả sử An và Bình trực tiếp trao khóa mã K cho nhau và "tin tưởng nhau" là không làm lộ khóa mã cho người thứ ba, như vậy khi nhận được thông điệp được mã hóa bởi K, hai đối tác có thể nhận dạng ra thông điệp đó chính là do đối tác của mình gửi. Mặt

khác nếu Bình nhận được thông điệp của An mã hóa bởi K thì An không thể chối bỏ rằng không phải do mình phát hành thông điệp đó (vì ngoài Bình chỉ có An biết khóa K). Tuy nhiên khi Bình nhận được mà chối là không nhận được thì phải do "tính tin tưởng" giữa hai đối tác chứ không phải do khóa K đảm bảo.

- Mã hóa đối xứng không đảm bảo tính toàn vẹn dữ liệu. Giả sử thư của An gửi cho Bình đã lọt vào tay Công. Công không hiểu gì về nội dung thông điệp nhưng vẫn có thể thêm bớt dữ liệu làm thay đổi, sai lệch nội dung thông điệp rồi vẫn gửi tiếp cho Bình: Bình không thể biết là thông điệp đã bị thay đổi nội dung (Có thể do không biết khóa mã nên dữ liệu thêm bớt của Công có thể làm cho thông điệp không giải mã được hay là vô nghĩa nhưng Bình vẫn không thể chắc chắn là có người can thiệp mà vẫn nghĩ là chính do An tạo ra như vậy!)

Vì những lý do trên các thuật toán mã hóa đối xứng loại này là những phương pháp mã hóa lý tưởng cho một người sử dụng (*single user*) với mục đích mã hóa dữ liệu của cá nhân hay tổ chức đơn lẻ để chống xâm nhập của kẻ xấu. Không phải chỉ có những bí mật về an ninh quốc phòng mà ngay những thông tin bí mật trong công nghệ, trong thương mại v.v. đều có thể là mục tiêu xâm nhập của những gián điệp công nghệ, kinh tế, hoặc xâm nhập trực tiếp hoặc sử dụng các biện pháp như gửi và cài Spyware, Trojan hay các phần mềm độc. Vì vậy cá nhân hay tổ chức, trước khi lưu giữ các dữ liệu thông tin quan trọng có thể và nên mã hóa bằng những khóa mã tự tạo và giữ bí mật khóa cho riêng mình biết.

Mã đối xứng bộc lộ hạn chế khi thông tin mật cần được chia sẻ với một bên thứ hai vì khi đó cần phải chuyển giao chìa khóa cho đối tác mà việc chuyển giao chìa khóa trong môi trường mở có nhiều nguy cơ bị lộ và như vậy việc mã hóa về sau trở thành vô nghĩa!

Mã đối xứng chỉ có thể sử dụng cho nhiều đối tác (*multiple users*) với điều kiện là có thể "mặt đối mặt" để trực tiếp chuyển giao khóa mã trong môi trường tin cậy hoặc có một biện pháp tin cậy nào

đó để chuyển giao khóa mã một cách an toàn. Nếu không có biện pháp chuyển giao khóa mã an toàn, tương đương với việc trao tay trực tiếp thì hầu như mã đối xứng không đảm bảo được yêu cầu nào trong 5 nguyên lý bảo mật thông tin đã nêu ở chương trước cả! (Vấn đề này sẽ được xem xét ở những chương 3 và 4 sau đây).

2.5. MỘT SỐ PHẦN MỀM MÃ HÓA ĐỐI XỨNG

Blowfish

Blowfish là một thuật toán mã hóa đối xứng (64 bit cipher) do Bruce Schneier thiết kế năm 1993. Blowfish có các độ dài khóa từ 32 đến 448 bit. Người ta đã nghiên cứu phân tích khá kỹ về các thuộc tính của Blowfish và nó cũng được đánh giá là một thuật toán mã hóa mạnh.

CAST

CAST được đặt theo tên viết tắt của các nhà phát minh ra nó là Carlisle Adams và Stafford Tavares. CAST là một thuật toán mã hóa rất phổ biến, mã hóa khối cipher 64 bit và cho phép độ dài khóa lên đến 128 bit.

IDEA

Thuật toán mã hóa dữ liệu quốc tế IDEA (International Data Encryption Algorithm) là một thuật toán mã hóa đối xứng do TS. X. Lai và GS. J. Massey xây dựng nhằm thay thế thuật toán DES chuẩn. IDEA cũng sử dụng khóa có độ dài là 128 bit. Kích thước lớn của khóa làm cho IDEA rất khó bị phá vỡ bằng tấn công bạo lực do thời gian duyệt tất cả các khả năng có thể có của khóa là quá lớn.

RC2

RC2 là một thuật toán mã hóa có kích thước khóa thay đổi. Ron Rivest đã thiết kế RC2 cho Công ty An toàn Dữ liệu RSA nhưng mọi chi tiết vẫn giữ bí mật, chưa được công bố.

RC4

RC4 cũng là một thuật toán do Ron Rivest phát triển năm 1987. Đấy là một thuật toán mã hóa dòng với khóa có kích thước thay đổi. Kích thước khóa của RC4 có thể đạt tới 2048 bit (thông thường là 256 bit)

RC6

RC6 là thuật toán mã hóa khối đối xứng do Ron Rivest, Matt Robshaw, Ray Sidney, và Yiqun Lisa Yin thiết kế nhằm đáp ứng yêu cầu của cuộc thi AES (Advanced Encryption Standard). Thuật toán RC6 là phần mềm lọt vào chung kết của cuộc thi đó và được chọn là phần mềm mã hóa tiên tiến tiêu chuẩn (*AES*).

Serpent

Serpent là thuật toán mã hóa khối đối xứng do Ross Anderson, Eli Biham and Lars Knudsen phát triển. Serpent có thể làm việc với nhiều tổ hợp khóa có độ dài khác nhau. Serpent cũng là một trong 5 phần mềm lọt vào chung kết cuộc thi AES.

Twofish

Twofish là một thuật toán mã hóa đối xứng khối, có kích thước khối là 128 và chấp nhận các khóa có mọi độ dài cho đến 256 bit. Twofish do Bruce Schneier, John Kelsey, Chris Hall, Niels Ferguson, David Wagner and Doug Whiting thiết kế. Viện quốc gia về tiêu chuẩn và công nghệ NIST (*The National Institute of Standards and Technology*) đã chấp nhận đầu tư để Twofish trở thành một trong các dự án thay thế cho thuật toán mã hóa DES trước đây.

3

QUẢN LÝ VÀ PHÂN PHỐI KHÓA

Như đã thấy ở chương 2, nhược điểm lớn nhất của mã hóa đối xứng là vấn đề chuyển giao, trao đổi khóa mã giữa các đối tác trong môi trường không tin cậy. Rõ ràng là một người dùng có thể sử dụng mã hóa đối xứng để bảo vệ rất tốt thông tin của chính mình chống sự xâm nhập của kẻ khác nhưng nếu muốn sử dụng được mã hóa đối xứng trong bảo mật thông tin giao dịch giữa nhiều đối tác thì nhất thiết phải xác lập những phương thức chuyển giao khóa mã an toàn.

3.1. TRUNG TÂM PHÂN PHỐI KHÓA (KDC)

3.1.1. Khái niệm KDC

Trong mật mã học, Trung tâm phân phối khóa (KDC: Key Distribution Center) là một phần của một hệ thống mật mã có mục đích giảm thiểu những hiểm họa khi trao đổi khóa mã giữa các đối tác. KDC thường được tổ chức thành hệ thống, trong đó một số người dùng có thể được phép sử dụng một vài dịch vụ chỉ trong một khoảng thời gian nào đó.

Chẳng hạn, một người quản trị mạng máy tính thiết lập một quy định chỉ cho phép một số người dùng được sử dụng chức năng phục hồi dữ liệu từ một số văn bản (có thể vì sợ rằng nếu để sử dụng tùy

tiện thì có những kẻ xấu sẽ thâm nhập được những thông tin nội bộ cần bảo mật). Nhiều hệ điều hành có thể kiểm tra việc tiếp cận chức năng phục hồi dữ liệu văn bản đó thông qua một "dịch vụ hệ thống". Nếu dịch vụ hệ thống đó được tổ chức theo cơ chế là chỉ cấp quyền truy cập chức năng cho những người dùng nào có một "thẻ chứng nhận quyền truy cập" thì vấn đề quy lại chỉ là việc tổ chức cấp thẻ cho những đối tượng mà hệ thống công nhận là thích hợp. Trong trường hợp thẻ chứng nhận đó là một khóa mã hoặc bao gồm một khóa mã thì ta có thể xem cơ chế đó như là một KDC.

3.1.2. Mô tả hoạt động

Hoạt động điển hình của các KDC gồm trước hết là việc tiếp nhận một yêu cầu của người dùng đối với một dịch vụ nào đấy. KDC dùng kỹ thuật mã hóa để nhận tính xác thực của dạng người dùng, tiếp đó kiểm tra xem người dùng đó có thuộc danh sách người được quyền sử dụng dịch vụ mà họ yêu cầu không. Nếu xác thực và kiểm tra đúng thì KDC có thể cấp thẻ chứng nhận truy cập.

KDC thường hoạt động với các mã khóa đối xứng.

Trong phần lớn các trường hợp KDC chia sẻ một khóa mã với mỗi đối tác. KDC tạo một thẻ chứng nhận dựa trên một khóa mã hóa máy chủ (*server key*). Người dùng nhận khóa mã đó và xuất trình cho máy chủ tương ứng kiểm tra, nếu phù hợp thì sẽ cấp quyền truy cập sử dụng dịch vụ.

3.2. TRAO ĐỔI KHÓA DIFFIE (D-H)

3.2.1. Khái niệm (D-H)

Trao đổi khóa D–H (Diffie–Hellman) là phương thức sử dụng một sơ đồ đặc biệt dùng để trao đổi khóa mã giữa các đối tác một cách an toàn. Phương thức Diffie–Hellman cho phép hai đối tác không biết gì với nhau từ trước có thể thỏa thuận với nhau để sử dụng chung một

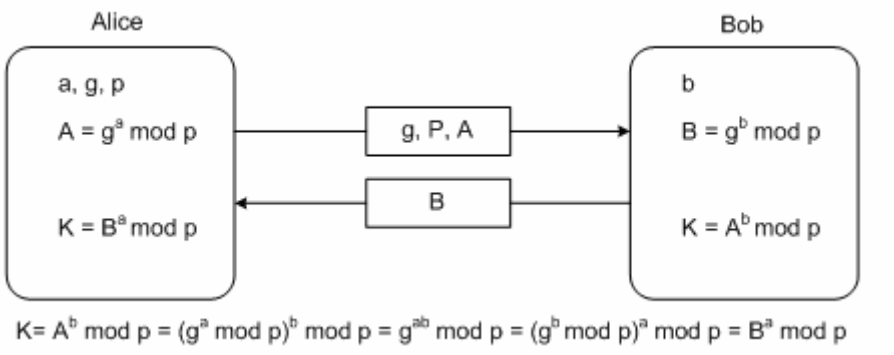
khóa mã bí mật thông qua một môi trường giao dịch không an toàn. Khóa bí mật đó (thường là một khóa đối xứng có tốc độ lập mã, giải mã nhanh chóng) về sau sẽ được hai hoặc nhiều đối tác sử dụng cho những thông điệp giao dịch nội bộ của mình.

Sơ đồ trao đổi khóa này được Whitfield Diffie và Martin Hellman công bố lần đầu tiên vào năm 1976 trong một công trình hợp tác nghiên cứu về phương thức chia sẻ bí mật qua một kênh truyền thông không tin cậy. Đến năm 2002, Hellman đề nghị gọi tên thuật toán là **trao đổi khóa Diffie-Hellman-Merkle** để ghi nhận đóng góp của Ralph Merkle. Tiếp đó, John Gill đề nghị ứng dụng thêm các bài toán logarit rời rạc, ý tưởng này đã được Malcolm Williamson nghiên cứu trước đó ít lâu nhưng mãi đến 1997 mới công bố công khai.

3.2.2. Mô tả

Diffie-Hellman đã thiết lập một sơ đồ trao đổi bí mật riêng tư có thể sử dụng cho việc truyền các thông tin bí mật bằng cách trao đổi dữ liệu qua một mạng truyền thông công cộng.

Sau đây là sơ đồ minh họa (hình 3.1).



Hình 3.1: Trao đổi khóa Diffie-Hellman

Ý tưởng đơn giản và độc đáo của thủ tục này là việc ứng dụng một nhóm nhân số tự nhiên modulo p, trong đó p là một số nguyên tố còn g là nguyên tố gốc mod p. Xem ví dụ sau đây:

An				**Bình**		
Bí mật	Công khai	Tính toán	Gửi	Tính toán	Công khai	Bí mật
a	p, g		p, g →			b
a	p, g, A	$g^a \bmod p = A$	A →		p, g	b
a	p, g, A		← B	$g^b \bmod p = B$	p, g, A, B	b
a, s	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, s

1. An và Bình thống nhất dùng số nguyên tố p=23 và cơ sở g = 5.
2. An chọn một số nguyên bí mật a = 6, rồi gửi cho Bình số A = g^a mod p (công khai)

   $A = 5^6 \bmod 23$

   $A = 15.625 \bmod 23$

   $A = 8$
3. Bình chọn một số nguyên bí mật b = 15, rồi gửi cho An số B = g^b mod p

   $B = 5^{15} \bmod 23$

   $B = 30.517.578.125 \bmod 23$

   $B = 19$
4. An tính toán: $s = B^a \bmod p$

   $s = 19^6 \bmod 23$

   $s = 47.045.881 \bmod 23$

   $s = 2$
5. Bình tính toán $s = A^b \bmod p$

   $s = 8^{15} \bmod 23$

   $s = 35.184.372.088.832 \bmod 23$

   $s = 2$

6. An và Bình chia sẻ nhau con số bí mật: s = 2. Sở dĩ như vậy là vì 6*15 cũng như là 15*6. Khi đó nếu bất kỳ người nào biết được cả hai số nguyên riêng của cả An và Bình thì cũng đều có thể tính được s như sau:

   $s = 5^{6*15} \bmod 23$

   $s = 5^{15*6} \bmod 23$

   $s = 5^{90} \bmod 23$

   $s = 807.793.566.946.316.088.741.610.050.849.573.099.185.363.389.551.639.556.884.765.625 \bmod 23$

   $s = 2$

Cả An và Bình cùng tìm ra một kết quả do bởi $(g^a)^b$ và $(g^b)^a$ là bằng nhau theo mod p. Chú ý là chỉ cần giữ bí mật a, b và $g^{ab} = g^{ba} \bmod p$. Mọi giá trị khác như p, g, $g^a \bmod p$, và $g^b \bmod p$ đều có thể gửi đi công khai. Một khi An và Bình đã tính ra được con số nguyên bí mật mà họ chia sẻ thì họ có thể sử dụng nó như là một khóa lập mã mà chỉ có hai người họ biết để trao đổi thông điệp cho nhau thông qua chính kênh thông tin mở đó. Tất nhiên nếu muốn đảm bảo bí mật hơn thì cần phải chọn các số a, b, và p khá lớn vì như ta thấy, có thể dễ dàng duyệt hết mọi giá trị có thể có của $g^{ab} \bmod 23$ vì rằng chỉ có 23 số nguyên có khả năng là số dư trong phép chia cho 23 (mod 23). Nếu p là một số nguyên tố với ít nhất khoảng 300 con số còn a và b có độ dài ít nhất 100 con số thì mọi thuật toán hiệu nghiệm nhất hiện nay được biết đến cũng đều không có khả năng tìm ra số a nếu chỉ biết các số g, p, $g^b \bmod p$ và $g^a \bmod p$, cho dù tận dụng mọi năng lực tính toán của con người. Bài toán này được biết đến dưới tên gọi là bài toán logarit rời rạc. Cũng nên chú ý thêm là g không cần phải chọn số lớn, trong thực hành người ta thường chỉ cần lấy g bằng 2 hoặc 5 là được.

Sau đây ta xem xét một cách mô tả tổng quát hơn của thuật toán

An và Bình thống nhất với nhau một *nhóm cyclic hữu hạn* G và một *phần tử sinh* g thuộc G. (Trong suốt toàn bộ thuật toán về sau ta

đều giả thiết như vậy và giả sử những kẻ tấn công đều biết được g). Ta sẽ viết nhóm G dưới dạng nhóm nhân.

1. Bình lấy một số tự nhiên bất kỳ b và gửi g^b cho An.

2. An tính $(g^b)^a$.

3. Bình tính $(g^a)^b$.

Bấy giờ cả Bình và An đều có phần tử g^{ab} của nhóm, phần tử đó có thể sử dụng như là khóa trao đổi giữa hai người. Các giá trị $(g^b)^a = (g^a)^b$ vì nhóm có tính kết hợp đối với phép nhân.

Sơ đồ tóm tắt: *(Giả sử tồn tại Công là một kẻ đọc lén thông tin giao dịch giữa An và Bình)*

Gọi s = khóa bí mật được chia sẻ. s = 2

g = cơ số công khai. g = 5

p = số nguyên tố công khai. p = 23

a = khóa bí mật của An. a = 6

A = khóa công khai của An. $A = g^a \bmod p = 8$

b = khóa bí mật của Bình. b = 15

B = khóa công khai của Bình. $B = g^b \bmod p = 19$

An		**Bình**		**Công**	
Biết	Không biết	Biết	Không biết	Biết	Không biết
p = 23	b = ?	p = 23	a = ?	p = 23	a = ?
Cơ số g = 5		Cơ số g = 5		Cơ số g = 5	b = ?
a = 6		b = 15			s = ?
$A = 5^6 \bmod 23 = 8$		$B = 5^{15} \bmod 23 = 19$		$A = 5^a \bmod 23 = 8$	
$B = 5^b \bmod 23 = 19$		$A = 5^a \bmod 23 = 8$		$B = 5^b \bmod 23 = 19$	
$s = 19^6 \bmod 23 = 2$		$s = 8^{15} \bmod 23 = 2$		$s = 19^a \bmod 23$	
$s = 8^b \bmod 23 = 2$		$s = 19^a \bmod 23 = 2$		$s = 8^b \bmod 23$	
$s = 19^6 \bmod 23$ $= 8^b \bmod 23$		$s = 8^{15} \bmod 23$ $= 19^a \bmod 23$		$s = 19^a \bmod 23$ $= 8^b \bmod 23$	
s = 2		s = 2			

3.2.3. Tính bảo mật

An rất khó có thể tính toán để tìm ra khóa riêng của Bình cũng như Bình khó tìm ra khóa riêng của An. Nếu điều đó dễ dàng thì kẻ đứng giữa Công có thể tấn công bằng cách gửi các khóa của mình giả mạo thay thế và có thể nắm bắt được mọi thông tin trao đổi giữa An và Bình đồng thời có thể gửi những thông điệp giả mạo.

Sau đây là lập luận của Diffie-Hellman để chứng tỏ điều đó (Chỉ sử dụng hai số bé để tiện cho thực hành).

Giao thức được xem là bí mật đối với những kẻ đọc lén nếu như G và g được chọn đúng đắn. Kẻ đọc lén phải giải bài toán Diffie-Hellman để phân tích được g^{ab}, điều này hiện nay được xem là rất khó. Một thuật toán giải được bài toán logarit rời rạc đó sẽ cho phép ta tính được a hoặc b và từ đó giải được bài toán Diffie-Hellman do đó làm cho thuật toán mã hóa này cũng như nhiều hệ thống mã hóa khóa công khai khác trở thành không an toàn nữa. Cấp của nhóm G phải là một số nguyên tố hoặc phải có một ước số nguyên tố lớn để không dùng được thuật toán Pohlig-Hellman khi tìm a hoặc b. Vì lý do đó đôi khi người ta dùng ***một số nguyên tố Sophie Germain*** q để tính $p=2q+1$, được gọi là ***số nguyên tố an toàn*** vì rằng cấp của G khi đó chỉ chia hết cho 2 và q. Lúc ấy nhiều khi ta thường chọn chính là g thay cho G để tổng quát hóa nhóm con cấp q của G, sao cho ***ký hiệu Legendre*** của g^a nhưng không bao giờ để lộ ra bit cấp thấp hơn của a.

Nếu An và Bình dùng những *số sinh ngẫu nhiên* có các số hệ quả không hoàn toàn ngẫu nhiên mà có thể dự đoán một mức độ nào đó thì công việc của kẻ nghe lén Công sẽ dễ dàng hơn nhiều. Các số nguyên bí mật a và b đều loại bỏ khi kết thúc phiên giao dịch. Vì vậy trao đổi khóa Diffie-Hellman có thể hướng tới khả năng bảo mật toàn vẹn vì không có khóa bí mật nào được tồn tại sử dụng lâu cho nên khả năng bị lộ khóa là rất thấp.

Trong mô tả đầu tiên, bản thân sơ đồ trao đổi của Difie-Hellman không cung cấp việc xác thực nhau của hai đối tác, do đó có khả năng bị sự tấn công của người đứng giữa. Một kẻ nghe lén như Công có thể tạo ra hai sự trao đổi Diffie-Hellman, lúc trao đổi với An thì mạo danh Bình và ngược lại lúc trao đổi với Bình thì mạo danh An, do vậy có thể tấn công nắm bắt được bí mật trao đổi của cả hai người. Vì vậy ta thấy nhất thiết cần phải có biện pháp xác thực đối tác khi sử dụng sơ đồ trao đổi khóa Diffie-Hellman.

3.2.4. Thỏa thuận khóa nhận dạng mật khẩu

Khi An và Bình chia sẻ một mật khẩu, họ phải dùng một dạng thỏa thuận khóa xác thực mật khẩu PAKE (Password-authenticated key agreement) của Diffie-Hellman để phòng ngừa tấn công của kẻ đứng giữa. Một sơ đồ đơn giản là dùng phần tử sinh g làm mật khẩu. Một đặc điểm của các sơ đồ này là một kẻ tấn công chỉ có thể thử một mật khẩu duy nhất cho một lần trao đổi với đối tác, do đó hệ thống có thể đảm bảo an toàn cao đối với cả những mật khẩu yếu. Sơ đồ này được mô tả trong bản Khuyến cáo X.1035 của ITU-T, sử dụng cho chuẩn kết nối mạng gia đình.

3.3. KERBEROS

Kerberos là một hệ thống giao thức xác thực an toàn trên mạng máy tính trước tiên được phát triển tại Viện Công nghệ Massachusett MIT (Massachusett Institute of Technology) về sau được dùng rộng rãi ở Mỹ. Kerberos cho phép các nút mạng chứng minh "căn cước" (*identity*) của mình với các đối tác, thông qua một môi trường giao dịch không tin cậy. Thoạt đầu Kerberos được thiết kế theo sơ đồ xác thực *client-server* (giữa máy khách - máy chủ) và sau đó cung cấp dịch vụ xác thực lẫn nhau (*mutual authentication*), Kerberos được mặc định sử dụng cổng 88.

Kerberos dùng mã khóa đối xứng. Các khóa đối xứng này được máy chủ Kerberos trao cho từng người sử dụng đã đăng nhập hệ thống. Mỗi người sử dụng được phép tạo một mật khẩu xác thực gửi cho máy chủ Kerberos để được nhận và dùng khóa mã. Để thực hiện được, hệ thống Kerberos đòi hỏi cấu hình mạng rất phức tạp và khó quản lý: máy chủ của mỗi website đăng ký sử dụng đều phải có máy chủ riêng được cài đặt Kerberos và máy chủ của hệ thống (bên thứ ba) sẽ phân phối khóa mã cho mỗi website đồng thời lưu giữ chúng lại để kiểm tra và đối chứng khi cần thiết.

3.3.1. Vài nét lịch sử

MIT phát triển Kerberos nhằm bảo vệ các dịch vụ mạng cung cấp bởi dự án có tên là **Athena**. Do vậy giao thức được đặt một tên gọi theo thần thoại Hy Lạp là Kerberos (hay *Cerberus*), tên của con chó ngao 3 đầu trấn giữ cung điện của Diêm vương Hades.

Hình 3.2: Diêm vương Hades và con chó ngao 3 đầu Cerberus

Các phiên bản 1 - 3 chỉ được lưu hành trong nội bộ MIT. Steve Miller và Clifford Neuman công bố phiên bản 4 vào cuối những năm

1980. Phiên bản 5 do John Kohn và C. Neuman thiết kế, được công bố năm 1993 lấy tên là RFC 1510 (sau đó nâng cấp thành RFC 4120 năm 2005) với ý đồ khắc phục những hạn chế về bảo mật của phiên bản 4.

Năm 2007 MIT thành lập công ty Kerberos nhằm phát triển thêm công cụ bảo mật này với sự bảo trợ của các công ty thương mại hàng đầu trong CNTT như Sun Microsystems. Apple Inc., Google, Microsoft… và một số trường đại học như Viện Công nghệ Hoàng gia KTH (KTH-Royal Institute of Technology) và Đại học Stanford.

Thoạt đầu chính phủ Hoa Kỳ cấm xuất khẩu Kerberos vì nó sử dụng thuật toán mã hóa DES với khóa 56 bit nên được xếp vào danh sách "công nghệ hỗ trợ quốc phòng". Tại Viện Công nghệ Hoàng gia KTH, Thụy Điển, dựa trên một phiên bản đơn giản eBones do MIT được phép xuất khẩu đã phát triển một phiên bản Kerberos bên ngoài Hoa Kỳ từ trước khi Hoa Kỳ thay đổi quy định về xuất khẩu mật mã năm 2000.

Windows 2000 và các hệ điều hành Windows tiếp sau đều sử dụng Kerberos làm phương thức xác thực mặc định. Một số phần bổ sung của Microsoft vào họ giao thức Kerberos suite được cung cấp trong phiên bản RFC 3244 (đặt và thay đổi mật khẩu) "*Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols*". RFC 4757 của Microsoft sử dụng thuật toán mã hóa RC4 cipher. Tuy nhiên Microsoft chỉ sử dụng giao thức Kerberos mà không sử dụng phần mềm được phát triển của MIT.

Nhiều hệ điều hành UNIX và tựa UNIX bao gồm FreeBSD, Mac OS X của hãng Apple, Red Hat Enterprise Linux 4, Sun's Solaris, AIX của IBM, OpenVMS của HP và nhiều hệ điều hành khác cũng tích hợp phần mềm xác thực Kerberos cho người dùng hay cho các dịch vụ của họ. Từ năm 2005, nhóm công tác IETF Kerberos liên tục cập

nhật những kết quả mới của họ. Những kết quả được cập nhật gần đây là:

- Encryption and Checksum Specifications" (RFC 3961).
- Advanced Encryption Standard (AES) Encryption for Kerberos 5 (RFC 3962).
- "The Kerberos Network Authentication Service (V5)" (RFC 4120)
- "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2." (RFC 4121).
- Mới nhất gần đây ngày 22 - 12 - 2010 - krb5-1.9 được công bố

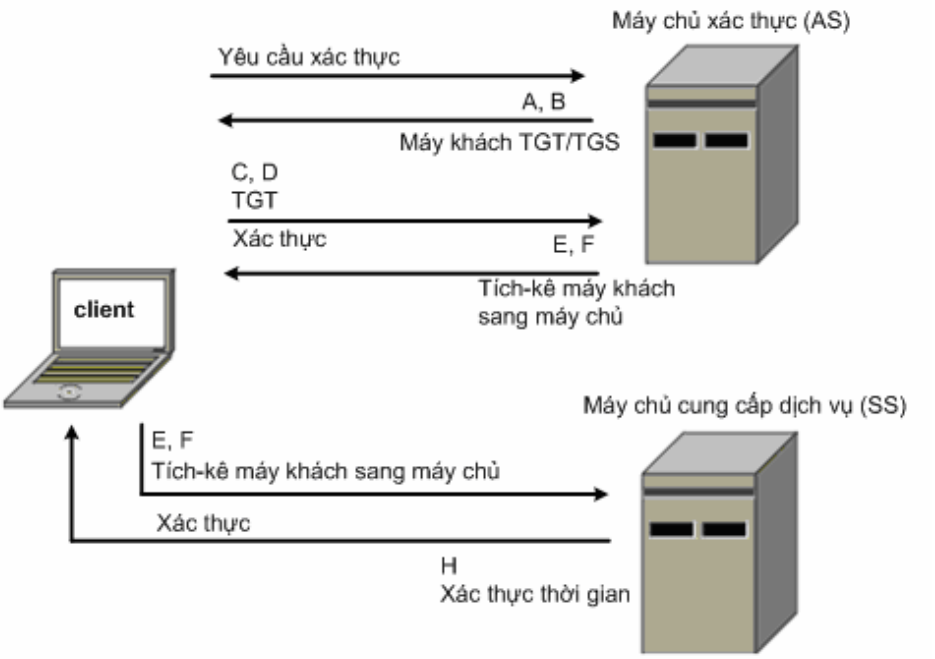
3.3.2. Cơ sở lý thuyết

Cơ sở lý thuyết của Kerberos là giao thức đối xứng Needham-Schroeder. Giao thức này sử dụng một bên thứ ba được tín nhiệm, chính là môt trung tâm phân phối khóa KDC (Key Distribution Center) gồm hai thành phần tách biệt nhau về mặt logic: Một máy chủ xác thực AS (Authentication Server), và một máy chủ cấp tích-kê TGS (Ticket Granting Server). Kerberos hoạt động dựa trên các "tích-kê" được sử dụng để xác thực căn cước của người dùng.

KDC lưu giữ một cơ sở dữ liệu khóa bí mật, mỗi thành viên trên mạng (tức là mỗi máy chủ hay người dùng bất kỳ) được chia sẻ một khóa mà chỉ có thành viên đó và KDC cùng biết mà thôi: khóa bí mật đó cũng dùng để chứng minh căn cước của thành viên. Khi hai thành viên cần giao tiếp, KDC sẽ sinh ra một khóa phiên nhằm bảo đảm tương tác giữa hai thành viên đó. Tính an toàn của giao thức phụ thuộc rất nhiều đến việc các thành viên đảm bảo giao dịch đồng bộ trong một thời gian ngắn thường gọi là tích-kê Kerberos.

3.3.3. Mô tả minh họa

Đầu phiên giao dịch, thành viên An được xác thực tại máy chủ xác thực (AS) và nhận được một "thẻ tích-kê" có đánh dấu thời gian. Tiếp đó An liên lạc với máy chủ cấp tích-kê (TGS) dùng thẻ tích-kê để chứng minh căn cước của mình và yêu cầu cung cấp dịch vụ. Nếu thẩm định đúng là An có quyền sử dụng dịch vụ đã yêu cầu thì TGS lại gửi thêm một tích-kê khác cho An. Bấy giờ An tiếp xúc với máy chủ cung cấp dịch vụ, xuất trình tích-kê mới để chứng minh rằng mình đã được cho phép sử dụng dịch vụ yêu cầu.



AS = Máy chủ xác thực;

SS = Máy chủ cung cấp dịch vụ;

TGS = Máy chủ cấp phát tích-kê;

TGT = Tích-kê cấp tích-kê.

Hình 3.3: Sơ đồ giao dịch thương lượng Kerberos

Thành viên được xác thực từ AS khi sử dụng một mật khẩu (bí mật chia sẻ dài hạn) và nhận được một TGT từ AS. Tiếp đó khi thành viên muốn tiếp xúc với SS nào thì người đó phải dùng tích-kê đó để đề nghị TGS cấp thêm cho mình một TGT bổ sung để giao dịch với SS mà không lộ mật khẩu bí mật đã chia sẻ giữa mình với AS. SS căn cứ vào TGT bổ sung để xác thực khách hàng để cung cấp dịch vụ.

Các bước chi tiết được mô tả như sau đây:

Đăng nhập phía khách:

1. Khách dùng một tên sử dụng và một mật khẩu (*username* & *password*) trên máy khách.
2. Khách thực hiện một hàm một chiều (thường là một hàm băm) đối với mật khẩu: đấy là khóa bí mật của khách /thành viên.

Nhận dạng khách:

1. Khách gửi một thông điệp rõ về ID của người sử dụng cho AS để yêu cầu dịch vụ (Chú ý: Không gửi khóa bí mật và mật khẩu cho AS). AS sinh một khóa bí mật bằng cách dùng hàm băm đối với mật khẩu của người dùng đã lưu ở cơ sở dữ liệu của mình (Active Directory trong máy chủ Windows)
2. AS kiểm tra xem người khách đã có trong cơ sở dữ liệu chưa. Nếu đã có, AS gửi lại 2 thông điệp cho khách:

+ Thông điệp A: *Khóa phiên TGS cho khách* được mã hóa bởi khóa bí mật của khách - người dùng.
+ Thông điệp B: *Tích-kê để nhận tích-kê* (*ticket-to Get-Ticket*) (bao gồm căn cước - ID của khách, địa chỉ mạng của khách và thời hạn có hiệu lực của tích-kê), được mã hóa bằng khóa bí mật của TGS.

3. Khi người khách nhận được 2 thông điệp A và B, phải giải mã thông điệp A bằng khóa bí mật sinh từ mật khẩu mà khách đã nhập. Nếu mật khẩu khách nhập không đúng với mật khẩu đã lưu trong cơ sở dữ liệu của AS thì khóa bí mật sẽ khác đi, do đó không giải mã được thông điệp. Nếu đúng, khách giải mã được thông điệp A và nhận được *khóa phiên TGS cho khách.* Khóa phiên này sẽ được dùng cho việc liên lạc về sau với TGS (Chú ý: Khách không thể giải mã thông điệp B vì thông điệp này mã hóa bằng khóa bí mật của TGS). Đến lúc đó, người khách đã có đủ thông tin để được xác thực bởi TGS.

Dịch vụ cấp phép:

1. Khi yêu cầu dịch vụ, khách gửi 2 thông điệp sau đây đến TGS: C

- Thông điệp C: Gồm TGT nhận được trong thông điệp B và ID của dịch vụ mình yêu cầu.

- Thông điệp D: Thông điệp xác thực (*Authenticator*) (gồm ID của khách và dấu xác nhận thời hạn hiệu lực), mã hóa bằng *khóa phiên TGS cho khách.*

2. Khi TGS nhận được 2 thông điệp C và D, sẽ tách thông điệp B từ trong C ra và dùng khóa bí mật của TGS để giải mã và thu được khóa phiên TGS cho khách. Dùng khóa phiên đó TGS giải mã thông điệp xác thực D và gửi cho khách 2 thông điệp sau đây:

- Thông điệp E: *Tích-kê máy khách sang máy chủ* gồm ID của khách, địa chỉ mạng của khách, thời hạn hiệu lực và *khóa phiên Máy khách/Máy chủ*) đều được mã hóa bởi khóa bí mật của máy chủ cung cấp dịch vụ khóa.

- Thông điệp F: *Khóa phiên máy khách/máy chủ* (*client/server*) được mã hóa bởi khóa phiên TGS cho máy khách.

Yêu cầu dịch vụ của khách hàng:

1. Khi nhận được thông điệp E và F từ TGS, khách hàng có đủ thông tin để được xác thực tại SS. Khách kết nối với SS để gửi 2 thông điệp sau:

- Thông điệp E của bước trước đây (*tích-kê máy khách đến máy chủ*, mã hóa bằng khóa bí mật của SS)
- Thông điệp G, một thông điệp xác thực khác, gồm ID của khách và thời hạn hiệu lực được mã hóa bằng *khóa phiên máy khách/máy chủ.*

2. SS giải mã tích-kê nhận được bằng khóa bí mật của bản thân để nhận được *khóa phiên máy khách/máy chủ.* Sử dụng khóa phiên đó, SS giải mã thông điệp xác thực và gửi lại thông điệp H (cũng dùng *khóa phiên máy khách/máy chủ* để mã hóa), nhằm khẳng định là SS đã xác thực được khách và sẵn sàng phục vụ (thời hạn hiệu lực được cộng thêm 1).

3. Khách giải mã H bằng *khóa phiên máy khách/máy chủ* và kiểm tra xem thời hạn hiệu lực đã cập nhật đúng chưa. Nếu đúng rồi thì có thể bắt đầu thực hiện dịch vụ mình yêu cầu.

4. SS cung cấp dịch vụ được yêu cầu cho khách (thanh toán, chuyển khoản, giao dịch khác,v.v.)

3.3.5. Một số nhược điểm

Kerberos có một số nhược điểm chính sau đây:

- Hệ thống yêu cầu phải có một máy chủ trung tâm hoạt động liên tục, nếu máy chủ Kerberos bị ngừng thì toàn hệ thống không còn truy cập được. Muốn tránh được điều này ta có thể sử dụng đồng thời nhiều máy chủ đồng dạng.

- Giao thức Kerberos yêu cầu thời gian đồng bộ rất chính xác vì các tích-kê dùng trong việc xác thực có một thời hạn hiệu lực rất ngắn. Nếu đồng hồ của các thành viên tham gia mạng có sai lệch đáng kể với đồng hồ của máy chủ Kerberos thì việc xác thực không thực hiện được do đó dịch vụ cũng không thể hoạt động.
- Giao thức quản trị không được tiêu chuẩn hóa giữa các cấu trúc mạng sử dụng khác nhau.
- Do mọi việc xác thực đều tập trung vào một KDC trung tâm nên nếu thiết bị trung tâm đó bị xâm nhập thì tất cả các thành viên sử dụng đều chịu ảnh hưởng.

4

MÃ HÓA KHÓA CÔNG KHAI

Như đã nói ở chương 2, các thuật toán mã hóa khóa đối xứng có một nhược điểm căn bản là hai người muốn trao đổi thông tin bí mật cần phải trao đổi khóa bí mật trước đó. Khóa bí mật này cần phải được trao đổi theo một cách thức an toàn, không phải bằng các phương thức thường dùng để liên lạc trong môi trường mở vì dễ bị lộ. Điều này khó thực hiện và nói chung là không thể đảm bảo bí mật, nhất là trong trường hợp muốn trao đổi thông tin với nhiều đối tác thì thực tế là không thực hiện được.

Vì vậy mã hóa khóa công khai (hay khóa bất đối xứng) được đưa ra như là một giải pháp thay thế. Thực ra mã bất đối xứng không thay thế hoàn toàn mã đối xứng mà người ta sử dụng đồng thời cả hai loại để bổ sung, hỗ trợ cho nhau.

4.1. VÀI NÉT LỊCH SỬ

Năm 1874, William Stanley Jevons xuất bản một cuốn sách mô tả mối quan hệ giữa các hàm một chiều (*one way function*) với mật mã học, đồng thời đi sâu vào bài toán phân tích ra thừa số nguyên tố (sử dụng trong thuật toán RSA). Tháng 7 năm 1996, một nhà nghiên cứu đã bình luận về cuốn sách trên như sau:

> *"Trong cuốn The Principles of Science: A Treatise on Logic and Scientific Method được xuất bản năm 1890, William S. Jevons đã phát hiện nhiều phép toán rất dễ thực hiện theo một chiều nhưng rất khó theo chiều ngược lại, điều đó chứng tỏ nhiều thuật toán mã hóa thực hiện rất dễ dàng trong khi giải mã thì rất khó khăn. Chẳng hạn tác giả nêu ra bài toán: ta có thể nhân để tìm tích số của các số nguyên tố nhưng ngược lại, muốn phân tích một số tự nhiên khá lớn ra các thừa số nguyên tố thì là điều không dễ dàng (thuật toán Euclide).*
>
> *Đây chính là nguyên tắc cơ bản của thuật toán* ***mật mã hóa khóa công khai*** *RSA (tuy rằng tác giả không phải là người phát minh ra mật mã hóa khóa công khai)"*

Thuật toán mật mã hóa khóa công khai được thiết kế lần đầu tiên bởi James H. Ellis, Clifford Cocks, và Malcolm Williamson tại Anh vào đầu thập kỷ 70 của thế kỷ trước. Thuật toán đó sau này được phát triển và biết đến dưới tên thuật toán Diffie-Hellman, và là một trường hợp đặc biệt của RSA. Tuy nhiên những thông tin này chỉ được tiết lộ ra vào năm 1997.

Năm 1976, Whitfield Diffie và Martin Hellman công bố một hệ thống mật mã hóa khóa bất đối xứng trong đó nêu ra phương pháp trao đổi khóa công khai. Công trình này chịu sự ảnh hưởng từ các công bố trước đó của Ralph Merkle về phân phối khóa công khai. Trao đổi khóa Diffie-Hellman là phương pháp đầu tiên có thể áp dụng trong thực tế để phân phối khóa bí mật trong môi trường mở, thông qua các kênh thông tin không an toàn. Kỹ thuật thỏa thuận khóa của Merkle có tên là hệ thống câu đố Merkle.

Thuật toán mã hóa khóa công khai có cơ sở hoàn chỉnh đầu tiên cũng được Ron Rivest, Adi Shamir và Leonard Adleman khởi xướng vào năm 1977 tại Học viện Kỹ thuật Massachusett MIT (Massachusett

Institute of Technology). Công trình này được công bố vào năm 1978 và thuật toán được đặt tên là thuật toán RSA - theo 3 chữ cái đầu của các đồng tác giả. RSA sử dụng phép toán lũy thừa theo modulo (với modulo được tính bằng tích số của 2 số nguyên tố lớn) để mã hóa và giải mã cũng như tạo chữ ký số. Độ an toàn của thuật toán được đảm bảo vì không tồn tại kỹ thuật hiệu quả để phân tích một số rất lớn thành thừa số nguyên tố.

Kể từ thập kỷ 1970, đã có rất nhiều thuật toán mã hóa, tạo chữ ký số, thỏa thuận khóa... được phát triển. Các thuật toán như ElGamal do Netscape phát triển hay thuật toán mã hóa đối xứng DSA do NSA và NIST chủ trì cũng dựa trên các bài toán lôgarit rời rạc tương tự như RSA. Vào giữa thập kỷ 1980, Neal Koblitz bắt đầu cho một dòng thuật toán mới: mật mã đường cong elliptic và cũng tạo ra nhiều thuật toán mã bất đối xứng. Mặc dù cơ sở toán học của dòng thuật toán này phức tạp hơn nhưng lại giúp làm giảm khối lượng tính toán đặc biệt khi khóa có độ dài lớn.

4.2. MÃ HÓA KHÓA CÔNG KHAI

Mã hóa khóa công khai là một dạng mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là *khóa công khai* (Public key) và *khóa riêng* (Private key) hay *khóa bí mật* (secret key).

4.2.1. Khái niệm chung

Thuật ngữ ***mã hóa bất đối xứng*** thường được dùng đồng nghĩa với ***mã hóa khóa công khai*** mặc dù hai khái niệm không hoàn toàn tương đương. Có những thuật toán mã bất đối xứng không có tính chất khóa công khai và bí mật như đề cập ở trên mà cả hai khóa (cho việc mã hóa và giải mã) đều cần phải giữ bí mật.

Trong mật mã khóa công khai, khóa riêng cần phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã.

Điều quan trọng đối với hệ thống là không thể (hoặc rất khó) tìm ra khóa bí mật nếu chỉ biết khóa công khai.

Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích:

- *Mã hóa:* giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
- *Tạo chữ ký số:* cho phép kiểm tra một văn bản xem nó có phải đã được tạo với một khóa bí mật nào đó hay không.
- *Thỏa thuận khóa:* cho phép thiết lập khóa để trao đổi thông tin mật giữa hai bên.

Thông thường, các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng do những ưu điểm nổi bật nên chúng được sử dụng nhiều.

Thuật toán mã hóa bất đối xứng sử dụng hai khóa: khóa công khai (hay khóa công cộng) và khóa bí mật (hay khóa riêng). Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã. Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người biết khóa công khai đều có thể mã hóa nhưng chỉ có người biết khóa riêng (bí mật) mới có thể giải mã được.

Ta có thể mô phỏng trực quan một hệ mã hóa khóa công khai như sau: Bình muốn gửi cho An một thông tin mật mà Bình muốn cho chỉ duy nhất An có thể đọc được. Để làm được điều này, An gửi cho Bình một chiếc hộp kín có ổ khóa đã mở sẵn và giữ lại chìa khóa. Bình nhận chiếc hộp, cho vào đó một lá thư viết bình thường

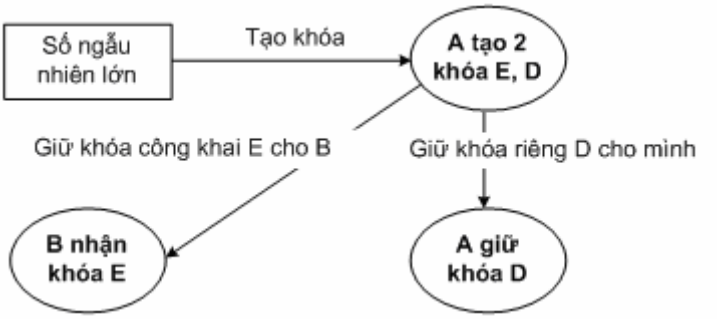
và bấm khóa lại (loại khóa thông thường chỉ bấm là khóa, sau khi sập chốt là khóa lại ngay cả Bình cũng không thể mở lại được, không đọc lại hay sửa thông tin trong thư được nữa). Sau đó Bình gửi chiếc hộp cho An qua bưu điện thông thường hoặc nhờ người nào đó mang hộ. Nhân viên bưu điện hay người mang hộ dù muốn cũng không thể mở hộp để xem thư. Chỉ khi chiếc hộp đến tay An, An có chìa khóa riêng mới mở được hộp và đọc được thông tin trong thư. Trong ví dụ này, chiếc hộp với ổ khóa mở An gửi cho Bình đóng vai trò khóa công khai, chiếc chìa khóa riêng của An chính là khóa bí mật.

4.2.2. Sơ đồ tạo và chuyển giao khóa công khai

Các hệ thống mã hóa khóa công khai thông thường được thực hiện với 3 bước cơ bản. Bước thứ nhất là công đoạn sinh khóa, một cặp khóa **public key** và **private key** có quan hệ về toán học được tạo ra dựa vào các bài toán cửa lật một chiều. Bước hai là bước mã hóa sử dụng khóa công khai (public key), khóa này có thể được chuyển giao trên môi trường mở. Quá trình giải mã là bước cuối cùng sử dụng khóa riêng bí mật (private key).

Các bước thực hiện như sau:

- A chọn một số ngẫu nhiên lớn để sinh cặp khóa, khóa công khai E và khóa bí mật riêng D.
- A gửi E-khóa công khai (public key) cho B, giữ D-khóa riêng (private key) cho mình.
- *Dùng khóa công khai để mã hóa, nhưng dùng khóa bí mật để giải mã.*
- B nhận được khóa công khai E. B có thông điệp gốc P, dùng E mã hóa E(P) = C, C là thông điệp mã hóa gửi cho A.
- A nhận được C, dùng D giải mã D(C) = P: được lại thông điệp gốc.



Hình 4.1: Chuyển giao khóa công khai

+ Chỉ riêng có A (có D) mới giải mã được

+ Ai có E đều mã hóa được

+ D dùng để giải E, nhưng nếu *chỉ biết* E thì hầu như chắc chắn *là không thể tìm được D.*

4.2.3. Phong bì số dạng đơn giản

Mã đối xứng có nhiều ưu điểm nhất là tốc độ lập mã và giải mã nhanh chóng. Thế nhưng nó lại có nhược điểm căn bản là sự không an toàn khi chuyển giao khóa trong môi trường không tin cậy. Ngược lại, mã bất đối xứng đảm bảo được an toàn trong việc chuyển giao khóa mã nhưng lại có nhược điểm là tốc độ lập mã, giải mã rất chậm.

Phong bì số (*Digital envelope*) là một biện pháp kết hợp của hai loại mã đối xứng và bất đối xứng để chuyển giao thông điệp an toàn và tin cậy. Trong trường hợp giao dịch 2 đối tác có thể dùng sơ đồ trao đổi khóa công khai nói trên làm một phong bì số để chuyển giao khóa mã đối xứng cho đối tác của mình trong môi trường giao dịch không tin cậy (chẳng hạn trong điều kiện không thể có "mặt đối mặt") như là dạng sau đây.

Sơ đồ chuyển giao khóa bí mật bằng phong bì số dạng đơn giản:

Bước 1: Tạo phong bì số

- A tạo khóa công khai E_1 gửi cho B, giữ khóa riêng D_1

- B tạo khóa riêng D_2 (của B) giữ cho mình, tạo khóa công khai E_2 (của B), dùng E_1 (nhận từ A) mã hóa: $E_1(E_2) = E'_2$ gửi E'_2 cho A.
- Chỉ có A sở hữu khóa riêng D_1 nên giải mã được: $E_1(E'_2) = E_2$. Từ đó *chỉ có A và B cùng sở hữu khóa công khai E_2 (do B tạo)*

Bước 2: Chuyển giao khóa đối xứng

- A tạo khóa đối xứng K dùng E_2 mã hóa: $E_2(K) = K'$ gửi cho B
- B dùng D_2 giải mã: $D_2(K') = K$
- Chỉ có A và B cùng biết khóa K, từ đó giao dịch bằng khóa đối xứng K.

Để tăng tính an toàn, A hoặc B thường xuyên có thể thay đổi khóa đối xứng và dùng phong bì số đã tạo để chuyển giao các khóa đối xứng mới cho nhau.

Tuy nhiên cần chú ý rằng phong bì số đơn giản loại này nếu sử dụng lâu thì có nhiều nguy cơ bị "tấn công của người đứng giữa" (man-in-the-middle attack) cho nên thông thường khi đã trao đổi xong một phong bì số đơn giản hai đối tác phải tiến hành "xác thực" lại bằng một phương pháp bổ sung nào đó.

4.2.4. Vấn đề phân phối khóa công khai

Cũng giống như các thuật toán mã hóa khác, cách thức phân phối khóa công khai là một trong những yếu tố quyết định đối với độ an toàn của mã bất đối xứng. Quá trình phân phối khóa cần chống lại được tấn công của người đứng giữa.

Giả sử người thứ ba Công có thể gửi cho Bình một khóa bất kỳ và khiến Bình tin rằng đó là khóa (công khai) của An. Như vậy đồng thời Công có khả năng đọc được thông tin trao đổi giữa Bình và An. Muốn vậy, Công sẽ gửi cho Bình khóa công khai của chính mình

(và làm cho Bình nghĩ rằng đó là khóa của An). Sau đó, Công đọc tất cả văn bản mã hóa do Bình gửi, giải mã với khóa bí mật của mình, giữ lại một bản copy đồng thời mã hóa bằng khóa công khai của An và gửi cho An. Về nguyên tắc, cả Bình và An đều không phát hiện ra sự can thiệp của người thứ ba. Các phương pháp chống lại dạng tấn công này dựa trên các chứng thực số (digital certificate) hoặc các thành phần của hạ tầng khóa công khai PKI (Public Key Infrastructure - xem chương 5).

4.3. THUẬT TOÁN RSA

Thuật toán này được Rivest, Shamir và Adleman mô tả lần đầu tiên năm 1977 tại trường Đại học MIT.

Giả sử An và Bình cần trao đổi thông tin bí mật thông qua một kênh không an toàn (ví dụ như qua Internet). Với thuật toán RSA, An đầu tiên cần tạo ra cho mình một cặp khóa gồm khóa công khai E và khóa bí mật D theo các bước sau:

4.3.1. Mô tả thuật toán

1. Chọn 2 *số nguyên tố* khá lớn (>1024bit) P và Q, $P \neq Q$
2. Lấy tích số: N = PQ, N được gọi là modulo mã hóa.
3. Chọn số E sao cho: $1 < E < PQ$, E và (P-1)(Q-1) nguyên tố cùng nhau (vậy E phải chọn là một số lẻ). E được gọi là số mũ mã hóa.
4. Tính số D sao cho tích số $DE \equiv 1 [\mathrm{mod}(P-1)(Q-1)]$ có nghĩa là tích số DE chia cho tích số (P-1)(Q-1) có số dư là 1, hay là DE-1 chia hết cho (P-1)(Q-1). Ta dùng phương pháp thử dần các số nguyên X sao cho có được: $D = [X(P-1)(Q-1) + 1]/E$ là số nguyên. D được gọi là số mũ giải mã.

Khóa công khai An gửi cho Bình (qua đường thông tin bất kỳ) là cặp số [N,E]

Khóa bí mật An giữ cho riêng mình là cặp số [N,D]

Mã hóa

- Bình nhận được khóa công khai của An gửi. Bình có thông điệp gốc (plaintext) T (thông điệp đã được số hóa, T thực ra là một con số dạng nhị phân được đổi thành số thập phân nào đó) cần gửi cho An.

- Bình mã hóa bằng phép toán: $T^E \bmod N = C$; T = plaintext, C = ciphertext. Phép toán "lũy thừa theo modulo" có nghĩa là *lấy T lũy thừa E rồi chia cho N và lấy số dư.*

- Bình gửi thông điệp mã hóa C cho An.

Giải mã

- An nhận được C.

- An giải mã bằng phép toán: $C^D \bmod N = T$.

- Như vậy là ở đây ta cần phải chứng minh được rằng:

$$(T^E \bmod N)^D \bmod N = T$$

Điều này đã được chứng minh bằng cách ứng dụng Định lý Trung Hoa về số dư (The Chinese Remainders Theorem) một thành tựu rất cao về số học, trong toán học Cổ Trung Hoa thường gọi là Bài toán Hàn Tín điểm binh (Hàn Tín là một vị tướng nhà Tiền Hán, vào khoảng thế kỷ thứ II trước công nguyên, xem phụ lục II). Thực chất việc tìm khóa riêng D chính là tìm một *phép toán ngược trong vành modulo N* của E.

Một số lưu ý:

- Các số nguyên tố thường được chọn bằng phương pháp thử ngẫu nhiên.

- Các bước 3 và 4 có thể được thực hiện bằng giải thuật Euclid mở rộng.

Một dạng khác của khóa bí mật:

- P và Q, hai số nguyên tố chọn ban đầu,
- D mod (P-1) và D mod (Q-1) (thường được ký hiệu là DmP1 và DmQ1),
- (1/Q) mod P (thường được gọi là iQmP)

Dạng này cho phép thực hiện giải mã và lập mã nhanh hơn với việc sử dụng định lý số dư Trung Hoa (Chinese Remainder Theorem) dạng CRT.

Ở dạng này, tất cả thành phần của khóa bí mật phải được giữ bí mật. An gửi khóa công khai cho Bình và giữ bí mật khóa riêng của mình.

Ở đây, P và Q giữ vai trò rất quan trọng. Chúng là các nhân tố của N và hỗ trợ cho khả năng tính D khi biết E.

Nếu không sử dụng dạng sau của khóa bí mật (dạng CRT) thì P và Q sẽ được xóa ngay sau khi thực hiện xong quá trình tạo khóa, chỉ giữ lại N, E, D.

Ví dụ: Ở đây chỉ để minh họa phương pháp nên ta chọn p, q khá bé cho dễ tính toán.

Chọn 2 số nguyên tố: p = 61 = $(111101)_2$; q = 53 = $(11011)_2$

(hủy ngay p và q sau khi tạo khóa),

n = p*q = 3233 - modulo

e = 17 - số mũ mã hóa (công bố công khai)

Khóa công khai A gửi đi cho B: (3233, 17)

d = 2753 - số mũ giải mã (A giữ riêng)

Thông điệp gốc (số hóa thành số dạng nhị phân rồi đổi ra số thập phân): 123

B dùng khóa công khai (n,e) mã hóa: $123^{17} \bmod 3233 = 855$

Thông điệp mã hóa được gửi đi: 855

A dùng khóa riêng (n,d) giải mã: $855^{2753} \bmod 3233 = 123$

4.3.2. Ưu và nhược điểm của mã RSA

Thuật toán *RSA* thực hiện một dãy phép tính lũy thừa modulo khá lớn.

Độ phức tạp tính toán

Khóa công khai = O(k2) bước tính toán, Khóa riêng = O(k3), Tổng quát mã RSA có độ phức tạp tính toán là O(k4) – k là số bit của modulo. Vì vậy mã RSA có nhược điểm đầu tiên là tốc độ lập mã và giải mã rất chậm.

Tuy nhiên mã RSA có độ bảo mật cao: hầu như không có thuật toán giải tổng quát mà phải dò thử dần (tấn công bạo lực). Nếu chọn P, Q lớn thì kết quả từ chỗ biết số mũ lập mã E, tìm ngược lại số mũ giải mã D rất phức tạp hầu như không làm được trong thời gian thực. Chẳng hạn ta tạo một khóa mã để mã hóa thông tin cho các thẻ tín dụng chỉ cho phép sử dụng trong 2 năm. Nếu khả năng bị phá khóa là trong thời gian 1000 năm hay lâu hơn nữa thì trong thực tế có thể xem là an toàn.

Một nhược điểm lớn khác của mã RSA là nguy cơ về "tính tin cậy". Khi B dùng khóa công khai nhận từ A để gửi tin, chắc chắn chỉ A đọc được: tin cậy phía người gửi tin. Khi A nhận tin, *chưa chắc do B gửi* (vì khóa công khai có thể lộ và người thứ ba biết khóa công khai, có thể dùng để mã hóa những thông điệp giả gửi cho A): không tin cậy phía người nhận tin.

Để khắc phục điều đó, phải có phương pháp "phân phối khóa công khai" một cách tin cậy hơn. Trong trường hợp chỉ có 2 đối tác trao đổi với nhau, người ta có thể dùng sơ đồ trao đổi khóa công khai để đảm bảo an toàn và tin cậy cho cả hai phía gửi và nhận tin.

Sơ đồ trao đổi khóa công khai

- A tạo một cặp khóa, khóa công khai (của A) là E_1 cho B và khóa riêng D_1 giữ cho mình.
- B tạo khóa riêng D_2, khóa công khai E_2 (của B).
- Dùng E_1 nhận được của A để mã hóa E_2: $E_1(E_2) = E'_2$, B gửi E'_2 cho A và giữ D_2 cho riêng mình.
- A nhận được E'_2, giải mã bằng D1 (Chỉ mình A có D_1): *Chỉ có A đọc được E_2.* Khi đó chỉ có 2 đối tác A và B cùng sở hữu khóa công khai E_2.
- A có thông điệp gốc P, dùng E_2 (của B mã hóa thông điệp: $E_2(P) = C$, gửi thông điệp mã hóa (bằng khóa công khai của B) cho *B chắc chắn chỉ có B đọc được.*
- B: nhận *chắc chắn do A gửi*, đọc: $D_2(C) = P$.

Sử dụng sơ đồ trao đổi khóa công khai, chúng ta tạo được sự tin cậy cả cho hai phía người gửi tin và người nhận tin. Nhưng mặt khác độ phức tạp tính toán tăng lên và tốc độ lập mã, giải mã càng chậm!

Mức độ an toàn

Về khía cạnh an toàn, các thuật toán mật mã hóa khóa bất đối xứng cũng không khác nhiều với các thuật toán mã hóa khóa đối xứng. Có những thuật toán được dùng rộng rãi, có thuật toán chủ yếu trên lý thuyết; có thuật toán vẫn còn được xem là an toàn, có thuật toán đã bị phá vỡ. Cũng cần lưu ý là những thuật toán được dùng rộng rãi không phải lúc nào cũng đảm bảo an toàn. Một số thuật toán có những chứng minh về độ an toàn với những tiêu chuẩn khác nhau. Nhiều chứng minh gắn việc phá vỡ thuật toán với những

bài toán nổi tiếng vẫn được cho là không có lời giải trong thời gian đa thức. Nhìn chung, chưa có thuật toán nào được chứng minh là an toàn tuyệt đối. Vì vậy, cũng giống như tất cả các thuật toán mật mã nói chung, các thuật toán mã hóa khóa công khai cần phải được sử dụng một cách thận trọng.

4.4. MỘT SỐ HỆ MẬT MÃ KHÓA CÔNG KHAI KHÁC

Trong mục này ta sẽ xem xét một số hệ mật mã khóa công khai khác.

Chẳng hạn như người ta cũng sử dụng **sơ đồ Diffie-Hellman** (Chương 3) như là một thuật toán tạo khóa công khai. Giả sử An và Bình đã thống nhất với nhau chọn một nhóm cyclic hữu hạn G_m, một phần tử sinh g thuộc G_m và p là một số nguyên tố công khai. An lấy một số nguyên bí mật cho riêng mình là a. Khóa công khai của An gửi cho Bình chính là $\mathbf{(g^a, g, p)}$.

Để gửi thông điệp của mình đến An, Bình chọn một số ngẫu nhiên b, và gửi g^b (không mã hóa) cho An cùng với thông điệp được mã hóa bởi khóa đối xứng $(g^a)^b$. Chỉ có An sở hữu a mới có thể giải mã thông điệp. Một khóa công khai được chia sẻ trước cũng có thể ngăn ngừa các tấn công của người đứng giữa.

Tuy nhiên trong thực tế thì ngày nay người ta không sử dụng khóa công khai theo sơ đồ Diffie-Hellman vì thuật toán mã hóa khóa công khai RSA là thuật toán được sử dụng quá phổ biến với các ưu điểm của nó và nhất là RSA đã thành lập được một cơ quan chứng thực điện tử hiện nay đang hoạt động rộng khắp chính là VeriSign.

Hệ mật mã Elgamal dựa trên bài toán logarit rời rạc cũng là một thuật toán được dùng khá phổ biến trong nhiều thủ tục mật mã. Ở các phần sau sẽ xem xét thêm đến một hệ mật mã khóa công khai ra đời sớm nhất là hệ mật mã xếp ba lô Merkle-Hellman và điểm qua sơ

lược một số hệ mật mã khóa công khai khác bao gồm các hệ thống loại Elgamal dựa trên các trường hữu hạn và các đường cong elliptic.

4.4.1. Hệ mật mã ElGamal

Hệ mật mã ElGamal là một thuật toán tương tự như hệ thống Diffie-Hellman trình bày ở mục sau, được xây dựng trên bài toán logarit rời rạc.

Dù rằng tác giả của hệ mật mã này (Taher Elgamal) không đăng ký xin cấp bản quyền cho sáng tạo của mình nhưng những người sở hữu bản quyền của hệ mật mã Diffie-Hellman vì lý do nào đó vẫn xem hệ này cũng thuộc phạm vi bảo vệ của giấy phép bản quyền của mình. Cũng không ai rõ lý do thực sự của việc đăng ký tên thuật toán là ElGamal (chữ G viết hoa) trong khi họ của tác giả là Elgamal (chữ g không viết hoa).

Có thể thấy ngay nhược điểm rõ ràng của hệ ElGamal là thông điệp sau khi mã hóa có kích thước rất lớn, xấp xỉ gấp hai lần thông điệp gốc! Chính vì vậy hệ mật mã này thường không dùng để mã hóa các khối dữ liệu thông tin lớn mà chủ yếu dùng cho các thông điệp ngắn chẳng hạn như để tạo các khóa chung.

Tạo khóa công khai ElGamal

Cũng như trong trường hợp của mã Diffie-Hellman, hai đối tác An và Bình có chung (công khai) một số nguyên tố p và một số sinh g (generator). An chọn một số ngẫu nhiên a và tính $A = g^a$, Bình cũng chọn một số ngẫu nhiên b và tính $B = g^b$. Khóa công khai của An là A và khóa riêng là a; tương tự như vậy, khóa công khai của Bình là B còn khóa riêng là b.

Mã hóa và giải mã thông điệp

Nếu Bình muốn gửi một thông điệp m cho An, Bình sẽ chọn ngẫu nhiên một số k bé hơn p rồi tính:

$c_1 = g^k \bmod p$; $c_2 = A^k * m \bmod p$ tiếp đó gửi c_1 và c_2 cho An.

An sử dụng c_1 và c_2 để tái hiện thông điệp bằng cách tính:

$c_1^{-a} * c_2 \bmod p = m$ bởi vì rằng:

$$c_1^{-a} * c_2 \bmod p = (g^k)^{-a} * A^k * m = g^{-a*k} * A^k * m$$
$$= (g^a)^{-k} * A^k * m = A^{-k} * A^k * m = 1 * m = m$$

4.4.2. Hệ mật mã "xếp ba lô" Merkle-Hellman

Mật mã xếp ba lô Merkle-Hellman là một trong những hệ mật mã khóa công khai ra đời sớm nhất, do Ralph Merkle và Martin Hellman phát minh vào năm 1978. Về mặt ý tưởng hệ mật mã này được xây dựng đơn giản hơn nhiều so với hệ RSA nhưng nó đã nhanh chóng bị đổ vỡ.

Mô tả

Merkle-Hellman là một hệ mật mã bất đối xứng, có nghĩa là khi giao dịch cần có hai khóa: một khóa công khai và một khóa riêng. Hơn nữa, cũng giống như RSA, hai khóa đó đều là một chiều với nghĩa là khóa công khai chỉ dùng để mã hóa còn khóa riêng chỉ dùng để giải mã. Cũng vì vậy nó không thể sử dụng để nhận dạng qua việc ký tên bằng mật mã.

Về mặt toán học, hệ Merkle-Hellman dựa trên bài toán tổng tập hợp con *subset sum problem* (một trường hợp riêng trong bài toán "cái ba lô" (*knapsack*) quen thuộc trong Toán rời rạc). Bài toán có thể phát biểu như sau: Cho một tập hợp các con số A và một con số b, hãy tìm một tập hợp con của A cộng lại bằng b. Trong trường hợp tổng quát, bài toán đó được biết là có tính NP- đủ (NP complete) *(khó giải bậc NP)*. Tuy nhiên trong trường hợp riêng khi tập hợp các con số (được gọi là cái ba lô) là "siêu tăng" (superincreasing) với nghĩa là có thể sắp xếp thành một dãy để cho mỗi phần tử của tập hợp đều lớn hơn tổng các phần tử đi trước nó, thì bài toán có thể

giải được "dễ dàng" trong thời gian đa thức bằng một thuật toán "tham lam" đơn giản.

Tạo khóa

Trong hệ mật mã Merkle-Hellman, các khóa là các "ba lô". Khóa công khai là một "ba lô đầy" còn khóa riêng là môt "ba lô vơi" (hard and easy knapsacks) kết hợp với hai số phần tử của phép cộng, một số nhân và một modulo, các số này được dùng để biến đổi các ba lô siêu tăng thành ba lô đầy. Những con số đó cũng được dùng để biến đổi tổng của các tập con của ba lô đầy thành tổng các tập con của ba lô vơi, tính toán thực hiện được trong thời gian đa thức.

Mã hóa

Để mã hóa một thông điệp, một tập con cả ba lô đầy được chọn ra bằng cách so sánh nó với một tập hợp các bit (plaintext) có độ dài bằng độ dài chìa khóa và làm cho mỗi thành phần ứng với số 1 trong plaintext một phần tử trong tập con mà bỏ qua những thành phần ứng với số 0 trong plaintext. Các phần tử của tập con đó cộng lại với nhau, tổng số thu được cho ta ciphertext.

Giải mã

Việc giải mã thực hiện được bởi vì số nhân và modulo đã dùng để biến đổi ba lô vơi siêu tăng thành khóa công khai, cũng có thể dùng để biến đổi con số đại diện cho ciphertext thành tổng các phần tử tương ứng của balô siêu tăng. Như vậy, dùng một thuật toán tham lam đơn giản, balô vơi giải ra được bằng cách dùng O(n) phép toán số học để giải mã.

Phương pháp toán học

Tạo khóa

Để mã hóa một thông điệp n bit, ta chọn một dãy siêu tăng:

$$w = (w_1, w_2, ..., w_n)$$

của n số tự nhiên khác 0. Lấy ngẫu nhiên một số nguyên q, sao cho:

$$q > \sum_{i=1}^{n} w_i$$

Và một số nguyên ngẫu nhiên r, sao cho USCLN(r,q) = 1 (r và q nguyên tố cùng nhau). q được chọn như vậy để bảo đảm cho ciphertext là duy nhất. Nếu chọn q bé hơn thì có thể có nhiều hơn một plaintext được mã hóa ra cùng một ciphertext. r phải nguyên tố cùng nhau với q nếu không sẽ không tồn tại số nghịch đảo mod q của nó. Sự tồn tại của số nghịch đảo là cần thiết cho quá trình giải mã thực hiện được.

Bây giờ ta tính dãy:

$$\beta = (\beta_1, \beta_2, ..., \beta_n)$$

trong đó:

$$\beta_i = r w_i \bmod q.$$

Khóa công khai chính là β, còn khóa riêng là: (w, q, r).

Mã hóa

Để mã hóa một thông điệp n bit: $\alpha \in (\alpha_1, \alpha_2, ..., \alpha_n)$,

trong đó α_i là bit thứ i của thông điệp và $\alpha_i \in \{0,1\}$, ta tính:

$$c > \sum_{i=1}^{n} \alpha_i \beta_i$$

Thông điệp mã hóa chính là c.

Giải mã

Để giải mã ciphertext c người nhận thông điệp cần tìm các bit α_i sao cho thỏa mãn:

$$c = \sum_{i=1}^{n} \alpha_i \beta_i$$

Đây là một bài toán rất khó nói chung nếu β_i là những giá trị bất kỳ vì người nhận thông điệp phải giải một loạt bài toán tổng tập hợp con mà bài toán đó đã được biết là NP- khó khăn! Tuy nhiên ở đây các giá trị của β_i đã được chọn sao cho việc giải mã là dễ dàng khi đã biết khóa riêng (w, q, r).

Mấu chốt của việc giải mã là phải tìm được một số nguyên s là *nghịch đảo của r theo modulo q*. Điều này có nghĩa là: s thỏa mãn phương trình:

$$s^*r \bmod q = 1$$

hay nói khác đi, tồn tại một số nguyên k sao cho:

$$sr = kq + 1.$$

Do bởi r đã được chọn sao cho USCLN(r,q) =1 chắc chắn có thể tìm được các số s và k bằng cách áp dụng thuật toán Euclid mở rộng. (Xem phụ lục II). Tiếp đó người nhận thông điệp tính:

$$c' \equiv cs(\bmod q)$$

Trong đó:

$$c' \equiv cs \equiv \sum_{i=1}^{n} \alpha_i \beta_i s(\bmod q)$$

Do bởi: $rs \bmod q = 1$ và $\beta_i = rw_i \bmod q$

kéo theo: $\beta_i s \equiv w_i rs \equiv w_i (\bmod q)$

Từ đó:

$$c' \equiv \sum_{i=1}^{n} \alpha_i w_i (\bmod q)$$

Tổng của mọi số w_i là bé hơn q do đó: $\sum_{i=1}^{n} \alpha_i w_i$ cũng vậy trong khoảng $[0, q-1]$.

Và như vậy người nhận phải giải bài toán tổng số các tập hợp con:

$$c' \equiv \sum_{i=1}^{n} \alpha_i w_i$$

Bài toán này giải được vì rằng w là một dãy siêu tăng. Chẳng hạn lấy phần tử lớn nhất trong w, gọi là w_k. Nếu $w_k > c'$, thì $\alpha_k = 0$, nếu $w_k \leq c'$, thì $\alpha_k = 1$. Sau đó trừ $w_k \times \alpha_k$ vào cho c', và lặp lại các bước cho đến khi tìm ra được mọi α_i.

Ví dụ: Cho một dãy siêu tăng:

w = {2, 7, 11, 21, 42, 89, 180, 354}

giả sử đấy là cơ sở của một khóa riêng, sử dụng nó để tính tổng:

$\sum$w = 706

Chọn một số q lớn hơn tổng số trên, chẳng hạn lấy: q = 881

Lại chọn một số r nằm trong khoảng [1,q) và nguyên tố cùng nhau với q: r = 588

Khóa riêng bây giờ gồm q, w và r.

Để tính ra một khóa công khai, hãy sinh một dãy β bằng cách nhân mỗi phần tử trong w với r mod q

β = {295, 592, 301, 14, 28, 353, 120, 236}

bởi vì:

2 * 588 mod 881 = 295

7 * 588 mod 881 = 592

11 * 588 mod 881 = 301

21 * 588 mod 881 = 14

42 * 588 mod 881 = 28

89 * 588 mod 881 = 353

180 * 588 mod 881 = 120

354 * 588 mod 881 = 236

Dãy β tạo nên khóa công khai.

Giả sử An muốn mã hóa thông điệp "a". Trước tiên An phải số hóa "a" thành một dãy ký tự {0,1} (dùng ASCII hoặc Unicode): a = 01100001

An nhân lần lượt mỗi bit với thành phần tương ứng trong β

a = 01100001

0 * 295

+ 1 * 592

+ 1 * 301

+ 0 * 14

+ 0 * 28

+ 0 * 353

+ 0 * 120

+ 1 * 236

= 1129

An gửi thông điệp mã hóa 1129 vào thùng thư và Bình nhận được. Để giải mã, Bình nhân 1129 với r^{-1} mod q (Xem nghịch đảo modulo ở phần phụ lục 2):

1129 * 442 mod 881 = 372

Bây giờ Bình phân tích số 372 thành những thành phần trong w bé hơn hoặc bằng 372, chọn từ các thành phần gần nhất cho đến khi còn số dư bằng 0:

372 - 354 = 18

18 - 11 = 7

7 - 7 = 0

Các thành phần ta chọn trong khóa riêng ứng với số 1 trong thông điệp gốc:

01100001

Biến đổi số nhị phân đó thành số thập phân, ta lại được "a".

4.4.3. Logarit rời rạc

Trong toán học, đặc biệt trong đại số học trừu tượng, logarit rời rạc (discrete logarithm) là một nhóm lý thuyết tương tự như logarit thông thường. Một logarit thông thường $\log_a(b)$ là nghiệm của phương trình $a^x = b$ trong trường số thực hay số phức. Tương tự như thế, nếu g và h là phần tử của một nhóm cyclic hữu hạn G thì một nghiệm của một phương trình $g^x = h$ cũng được gọi là logarit cơ số g của h trong nhóm G.

Ví dụ. Xét nhóm $(Z_p)^\times$ là tập hợp các lớp tương đương $\{1, \ldots, p - 1\}$ đối với phép nhân theo modulo của một số nguyên tố p. Nếu muốn tìm lũy thừa k của một số trong nhóm đó ta chỉ việc lấy lũy thừa k của số nguyên đó rồi chia kết quả thu được cho p và lấy số dư. Phép toán đó gọi là phép lũy thừa rời rạc hay lũy thừa modulo.

Chẳng hạn xét nhóm $(Z_{17})^\times$.

Muốn tính 3^4 trong nhóm đó, hoặc $3^4 \pmod{17}$ trước tiên ta tính $3^4 = 81$, chia 81 cho 17 được: 81 = 17 * 4 + 13: số dư là 13, vậy: $3^4 \pmod{17} = 13$. Logarit rời rạc là một phép toán ngược.

Chẳng hạn lấy phương trình $3^k \equiv 13 \pmod{17}$ đối với k. Như ta thấy trên phần đầu của ví dụ thì $k = 4$ là một nghiệm của phương trình nhưng không phải là nghiệm duy nhất.

Vì rằng do $3^{4+16n} \equiv 13 \times 1^n \equiv 13 \pmod{17}$, nếu n là một số nguyên thì: $3^{4+16n} \equiv 13 \times 1^n \equiv 13 \pmod{17}$ và như vậy phương trình này có vô số nghiệm dưới dạng $4 + 16n$.

Ngoài ra, vì 16 là số nguyên dương m bé nhất thỏa mãn phương trình $3^m \equiv 1 \pmod{17}$, ta gọi 16 là cấp (*order*) của 3 trong $(Z_{17})^\times$, thì khi đó lại chỉ có nghiệm duy nhất.

Tương tự như vậy đáp số có thể được biểu diễn là $k^6 \equiv 4 \pmod{16}$.

Định nghĩa

Tổng quát, cho G là một nhóm cyclic với *n phần tử*. Ta giả thiết đấy là nhóm nhân. Gọi b là một phần tử sinh của G, khi đó mọi phần tử g của G đều được viết thành dạng $g = b^k$ với k là một số nguyên nào đó. Nếu một cặp số nguyên như k_1 và k_2 cùng biểu diễn được thành $g = b^{k1} = b^{k2}$ thì k_1 và k_2 là đồng dư theo modulo n. Ta định nghĩa một hàm số:

$$\log_b : G \to \mathbb{Z}_n$$

(trong đó Z_n biểu thị vành số nguyên modulo n) bằng cách gán cho mỗi phần tử g một lớp đồng dư của k modulo n. Hàm số đó là một nhóm đẳng cấu gọi là logarit rời rạc cơ số b. Công thức đổi cơ số logarit thông thường vẫn đúng, chẳng hạn như nếu c là một phần tử sinh khác của G thì:

$$\log_c(g) = \log_c(b).\log_b(g)$$

Thuật toán

Hiện nay chưa có một thuật toán có hiệu lực nào được biết để tính toán logarit rời rạc $\log_b(g)$ tổng quát. *Thuật toán nhân tầm thường* là cứ nâng số b lên lũy thừa k cao mãi cho đến khi tìm được g. Thuật toán này đòi hỏi thời gian thực hiện tuyến tính đối với kích

thước của nhóm g tức là thời gian hàm mũ đối với số con số trong kích thước của nhóm G. Cũng có một số thuật toán nhanh hơn thuật toán nhân tầm thường nhưng cũng không có thuật toán nào thực hiện được trong thời gian đa thức (đối với số con số trong kích thước của nhóm). Có thể liệt kê ra đây một số thuật toán:

- Bước trẻ con và bước khổng lồ (Baby-step giant-step).
- Thuật toán Pollard cho lô-ga-rit (Pollard's algorithm for logarithms).
- Thuật toán chuột túi của Pollard (thuật toán Lamda – λ): Pollard's kangaroo algorithm (Pollard's lambda algorithm).
- Thuật toán Pohlig-Hellman
- Thuật toán tính chỉ số (Index calculus algorithm)
- Sàng lọc trường số (Number field sieve)
- Sàng lọc trường hàm số (Function field sieve)

Tuy rằng bài toán logarit rời rạc và bài toán thừa số nguyên là những bài toán hoàn toàn khác biệt nhưng chúng có chung một số tính chất sau:

- Cả hai bài toán đều rất khó (hiện nay chưa có một thuật toán giải hữu hiệu nào đối với các máy tính phi lượng tử, không phải là các "máy tính lượng tử").
- Cả hai bài toán đều được biết là có những thuật toán hữu hiệu trên máy tính lượng tử.
- Các thuật toán dùng được cho bài toán này thường cũng đều dùng được cho bài toán kia.
- Độ khó của cả hai bài toán đã được sử dụng để xây dựng một số hệ mật mã.

4.4.4. Hệ mật mã đường cong Elliptic

Mật mã đường cong elliptic ECC (Elliptic Curve Cryptography) là một dạng mã hóa khóa công khai dựa trên cấu trúc đại số của các đường cong elliptic trên những trường hữu hạn. Việc sử dụng các đường cong elliptic trong mật mã học do Neal Koblitz và Victor S, Miller đề xuất vào năm 1985. Trong giao dịch xã hội nói chung ít sử dụng hệ mật mã đường cong elliptic. Vì vậy trong phạm vi cuốn sách này chúng ta không đi sâu mô tả hệ thống mật mã này mà chỉ giới thiệu qua một số tính chất và đặc điểm của nó.

Mật mã khóa công khai dựa trên tính chất khó giải của một số bài toán tìm thuật toán ngược. Chẳng hạn như tính bảo mật của thuật toán RSA được bảo đảm là do tính chất khó khăn của bài toán phân tích một số tự nhiên lớn thành 2 hoặc nhiều thừa số nguyên tố. Đối với các thủ tục lập - giải mã dựa trên cơ sở đường cong elliptic thì có thể khẳng định rằng việc tìm được logarit rời rạc của một phần tử của đường cong elliptic ngẫu nhiên dựa trên một điểm cơ sở đã biết là không thể làm được. Kích thước của đường cong elliptic xác định độ khó của bài toán. Người ta tin rằng độ bảo mật của một hệ thống mã hóa RSA với modulo lớn có thể đạt được với một nhóm đường cong elliptic bé hơn rất nhiều. Mà nếu ta sử dụng một nhóm bé thì có thể giảm bớt bộ lưu trữ cũng như giảm bớt các yêu cầu về truyền tin.

Với các đối tượng của mật mã học hiện tại, một *đường cong elliptic* là một đường cong phẳng chứa những điểm có tọa độ thỏa mãn phương trình:

$$y^2 = x^3 + ax + b$$

Cùng với một điểm đặc biệt được gọi là điểm vô tận ký hiệu là ∞. Ở đây các tọa độ được chọn trong một trường hữu hạn cố định, có số đặc trưng khác 2 và 3 (nếu không, phương trình của đường cong sẽ có thể phức tạp hơn nhiều). Tập hợp đó cùng với phép toán nhóm

của lý thuyết nhóm elliptic lập thành một nhóm Abel (nhóm giao hoán) với điểm vô tận là phần tử trung hòa (phần tử đồng nhất: đơn vị θ).

Tính bảo mật trong ECC phụ thuộc vào khả năng tính được một điểm nhân nhưng không tính được thừa số nếu cho biết điểm gốc và điểm tích số.

Các sơ đồ mật mã

Một số thủ tục mã hóa dựa trên cơ sở logarit rời rạc được làm thích hợp với các thuật toán dựa trên cơ sở đường cong elliptic bằng các thay thế nhóm $(\square_p)^\times$ bởi một đường cong elliptic.

- Sơ đồ thỏa thuận khóa đường cong elliptic dựa trên sơ đồ Diffie–Hellman.
- Thuật toán Chữ ký số đường cong elliptic dựa trên thuật toán chữ ký số.
- Sơ đồ thỏa thuận khóa ECMQV dựa trên sơ đồ thỏa thuận khóa MQV.

Tại Hội thảo RSA năm 2005, NSA (Cơ quan bảo mật quốc gia Hoa Kỳ) đã công bố dãy B (suite B) một dãy thuật toán mật mã đặc biệt chỉ dùng ECC cho việc sinh chữ ký điện tử và trao đổi khóa. Dãy B nhằm sử dụng để bảo vệ cả hai loại thông tin và hệ thống được xếp hạng và không được xếp hạng bí mật cấp quốc gia.

5

CHỮ KÝ ĐIỆN TỬ VÀ CHỨNG THỰC ĐIỆN TỬ

5.1. KHÁI NIỆM VỀ CHỮ KÝ ĐIỆN TỬ

Trong một giao dịch, An gửi cho Bình một lá thư của mình. Việc gửi lá thư đó trước hết phải đảm bảo ba yêu cầu sau đây trong các nguyên lý bảo mật thông tin:

- *Tính bảo mật:* Lá thư dù lọt vào tay kẻ khác ngoài Bình thì kẻ đó cũng không hiểu được nội dung thư.
- *Tính toàn vẹn thông tin:* Nếu lá thư bị người trung gian làm biến đổi nội dung trong quá trình truyền tin thì Bình phải nhận biết là thư đã bị can thiệp (chỉ phát hiện (*detect*) nhưng có thể không biết nội dung bị can thiệp như thế nào để đính chính lại cho đúng (*correct*)).
- *Tính nhận biết:* Khi nhận được thư, Bình nhận ra được đúng là thư do An gửi, không phải là do một kẻ thứ ba giả mạo.
- *Tính không chối bỏ:* Sau này An không thể chối bỏ rằng lá thư đó không phải của mình.

Trong giao dịch thông thường, An ký tên vào lá thư để xác nhận rằng thư đó do mình phát hành, sau này không thể chối bỏ được. Khi Bình thấy chữ ký của An ở cuối thư thì tin tưởng là thư của An.

Trong giao dịch điện tử, nếu giữa An và Bình đã có sự trao đổi thống nhất một khóa mã bí mật K (chỉ hai người biết) thì nếu lá thư được mã hóa bằng khóa mã đó, hai yêu cầu nói trên đều thỏa mãn.

Tuy nhiên trong nhiều trường hợp, nếu có một thông điệp rất lớn cần gửi đi (Hợp đồng, cung cấp tư liệu v.v.) mà nội dung không có gì cần thiết phải bí mật toàn bộ, nếu phải mã hóa (và giải mã) thì quá phiền phức và tốn thời gian.

Vậy có cách nào giải quyết được bốn yêu cầu nói trên mà không cần phải mã hóa toàn bộ thông điệp không?

Nói cách khác, có thể tạo ra một công cụ đóng vai trò như chữ ký của người phát hành thông điệp trong dạng giao dịch thông thường không?

5.1.1. Chữ ký điện tử

Chữ ký điện tử (Electronic signature) chính là công cụ đáp ứng được những yêu cầu đề ra trên đây cho việc trao đổi thông điệp điện tử. Không những thế, ngoài ra chữ ký điện tử còn có một số tính chất khác đảm bảo các nguyên lý khác của vấn đề bảo mật dữ liệu như tính toàn vẹn thông tin, tính xác thực và tính nhận dạng đối tác.

Hiện nay, chữ ký điện tử có thể bao hàm các cam kết gửi bằng E-mail, việc nhập các số nhận dạng cá nhân (PIN) vào các máy ATM, ký bằng bút điện tử với thiết bị màn hình cảm ứng tại các quầy tính tiền[1], chấp nhận các điều khoản người dùng (EULA) khi cài đặt phần mềm máy tính, ký các hợp đồng điện tử online...

5.1.2. Các định nghĩa pháp lý

Nhiều luật được ban hành trên thế giới công nhận giá trị pháp lý của chữ ký điện tử nhằm thúc đẩy các giao dịch điện tử xuyên quốc gia.

Luật Giao dịch điện tử Việt Nam, Điều 4 định nghĩa:

(1) Chứng thư điện tử là thông điệp dữ liệu do tổ chức cung cấp dịch vụ chứng thực chữ ký điện tử phát hành nhằm xác nhận cơ quan, tổ chức, cá nhân được chứng thực là người ký chữ ký điện tử.

(2) Chứng thực chữ ký điện tử là việc xác nhận cơ quan, tổ chức, cá nhân được chứng thực là người ký chữ ký điện tử.

(5) Dữ liệu là thông tin dưới dạng ký hiệu, chữ viết, chữ số, hình ảnh, âm thanh hoặc dạng tương tự.

(12) Thông điệp dữ liệu là thông tin được tạo ra, được gửi đi, được nhận và được lưu trữ bằng phương tiện điện tử.

Bộ luật ESIGN (Hoa Kỳ), Điều 106 định nghĩa:

(2) Điện tử (*electronic*): chỉ các công nghệ liên quan tới điện, số, từ, truyền tin không dây, quang, điện từ hoặc các khả năng tương tự.

(4) Văn bản điện tử (*electronic record*): Các hợp đồng hoặc các văn bản khác được tạo ra, lưu trữ, trao đổi dưới dạng điện tử.

(5) Chữ ký điện tử (*electronic signature*): Các tín hiệu âm thanh, ký hiệu, quá trình gắn (*vật lý hoặc logic*) với hợp đồng hay văn bản và được thực hiện bởi người muốn ký vào hợp đồng hay văn bản đó.

Bộ luật GPEA (Hoa Kỳ), Điều 1710 định nghĩa:

(1) Chữ ký điện tử (*electronic signature*): là cách ký các văn bản điện tử đảm bảo:

(A) Nhận dạng và xác thực cá nhân đã tạo ra văn bản;

(B) Chỉ ra sự chấp thuận của người ký đối với nội dung trong văn bản.

Bộ luật UETA (Hoa Kỳ), Điều 2 định nghĩa:

(5) Điện tử (electronic 'valeking132') chỉ các công nghệ liên quan tới điện, số, từ, không dây, quang, điện từ hoặc các khả năng tương tự.

(6) Tác tử điện tử (*electronic agent*) là các chương trình máy tính hoặc các phương tiện tự động khác sử dụng độc lập để khởi đầu một hành động hoặc đáp lại các tín hiệu điện tử mà không cần sự giám sát của con người.

(7) Văn bản điện tử (electronic record 'valeking132') Các văn bản được tạo ra, lưu trữ, trao đổi dưới dạng điện tử.

(8) Chữ ký điện tử (*electronic signature*) Các tín hiệu âm thanh, ký hiệu, quá trình gắn (vật lý hoặc logic) với hợp đồng hay văn bản và được thực hiện bởi người muốn ký vào hợp đồng hay văn bản đó.

Commodity Futures Trading Commission 17 CFR Phần 1 Điều 1.3 định nghĩa:

(tt) Chữ ký điện tử là tín hiệu âm thanh, ký hiệu, quá trình gắn (vật lý hoặc logic) với hợp đồng hay văn bản và được thực hiện bởi người muốn ký vào hợp đồng hay văn bản đó.

Food and Drug Administration 21 CFR Điều 11.3 định nghĩa:

(5) Chữ ký số là các chữ ký điện tử dựa trên các phương pháp mật mã để nhận thực người tạo văn bản dựa trên các quy tắc và tham số sao cho có thể kiểm tra được nhận dạng của người tạo và tính toàn vẹn của văn bản.

(7) Chữ ký điện tử là các số liệu (máy tính) được tạo ra, chấp nhận và cho phép bởi cá nhân có thẩm quyền (tương đương với chữ ký văn bản giấy truyền thống).

Luật chữ ký điện tử của Trung Quốc

Mục tiêu hướng tới thống nhất việc thực hiện, khẳng định tính pháp lý và bảo vệ quyền lợi hợp pháp của các bên liên quan tới việc thực hiện chữ ký điện tử.

Liên minh châu Âu (EU)

Liên minh châu Âu (EU) đã thiết lập khung pháp lý cho chữ ký điện tử:

Hướng dẫn số 1999/93/EC của Quốc hội châu Âu ngày 13 tháng 12 năm 1999 về khung pháp lý của chữ ký điện tử.

Quyết định 2003/511/EC sử dụng 3 thỏa thuận tại hội thảo CEN làm tiêu chuẩn kỹ thuật.

Một số quốc gia đã thực hiện quyết định 1999/93/EC.

Áo: Luật Chữ ký, 2000

Anh, Scotland và Wales: Luật Thông tin điện tử, 2000

Đức: Luật Chữ ký, 2001

Li-thu-a-ni-a: Luật Chữ ký điện tử, 2002

Na Uy: Luật Chữ ký điện tử, 2001

Tây Ban Nha: Đạo luật 59/2003 ngày 19/12 về Chữ ký điện tử.

Thụy Điển: Đạo luật Chữ ký điện tử (SFS 2000:832).

Ấn Độ: Luật Công nghệ thông tin, 2000

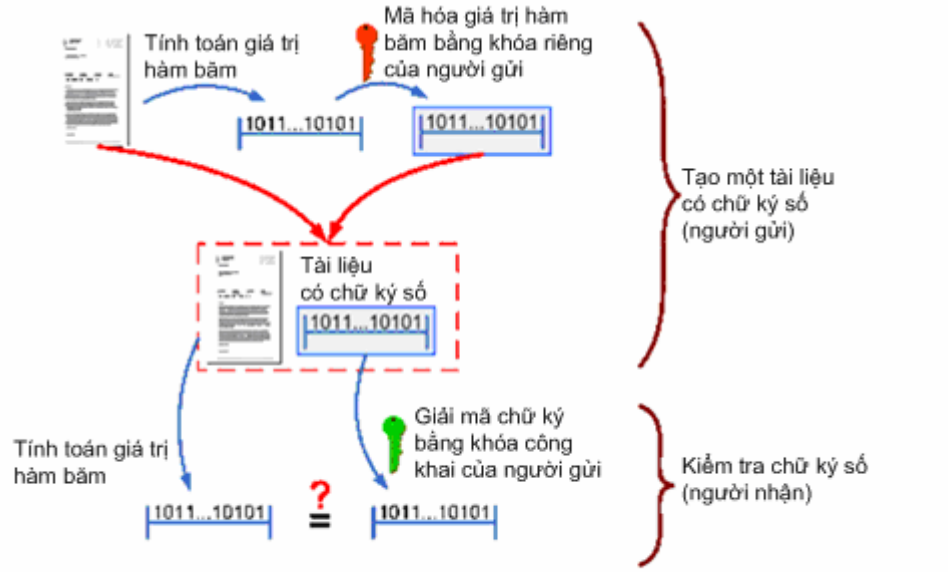
Niu Di-lân: Luật Giao dịch điện tử, 2003 Điều 22-24

5.1.3. Tạo chữ ký điện tử

Giả sử giữa An và Bình đã có trao đổi (riêng) một khóa mã K. Nếu An gửi cho Bình một thông điệp sử dụng khóa K để mã hóa thì chắc chắn Bình nhận ra thông điệp là do An phát hành, mặt khác vì chỉ có An và Bình cùng sở hữu khóa K nên An không thể chối là thông điệp không phải do mình tạo ra. Tuy nhiên nếu mục đích là để cho Bình nhận biết là mình đã dùng khóa K thì An không cần mã hóa toàn bộ thông điệp mà chỉ cần mã hóa một phần rất nhỏ của thông điệp rồi gửi cho Bình là đủ!

Chữ ký điện tử là một bộ phận thường có kích thước nhỏ tạo ra từ thông điệp, được người gửi mã hóa bằng khóa K đã trao đổi thống nhất giữa hai đối tác gửi và nhận thông điệp, gửi kèm với toàn bộ thông điệp cho người nhận.

Để tạo ra một bộ phận của thông điệp người ta thường dùng kỹ thuật hàm băm (*hash function*). Như vậy điều quan trọng ở đây không phải là Bình hiểu được "nội dung" của bộ phận thông điệp mã hóa là gì mà chủ yếu là nhận ra đối tác của mình bằng quy luật mã hóa đã trao đổi thống nhất. Chữ ký của cùng một người gửi, kèm vào trong các thông điệp khác nhau do người đó phát hành có thể có nội dung hoàn toàn nhau, điều quan trọng duy nhất là quy luật mã hóa K vẫn giữ nguyên!



Nếu mã hàm băm tính được không trùng với kết quả giải mã chữ ký số thì kết luận là nội dung tài liệu nhận được đã bị sửa đổi so với tài liệu gốc của người gửi

Hình 5.1: Sơ đồ tạo và kiểm tra chữ ký điện tử

5.1.4. Chữ ký số

Việc tạo chữ ký điện tử qua mã hóa giá trị băm của mỗi thông điệp quả thực không dễ dàng với những cá nhân, tổ chức không được trang bị tốt về công nghệ thông tin. Nếu chỉ nhằm mục đích nhận biết và không chối bỏ, người ta thường dùng một phương pháp đơn giản hơn: đó là các chữ ký số (*digital signature*).

Chữ ký số có thể xem là một lớp con của chữ ký điện tử. Sau khi hai đối tác đã trao đổi khóa mã K, An dùng khóa K để mã hóa một nội dung dữ liệu cố định S nào đó: K(S) = S' và sẽ gắn S' vào mọi thông điệp của mình phát hành. Khi An nhận được một thông điệp có gắn S', dùng K để giải mã được lại S thì nhận ra thông điệp là do An phát hành. S' là chữ ký số của An. Nội dung chữ ký số rất phong phú: có thể là một đoạn văn bản (họ và tên, chữ ký thật scan lên máy tính, chữ ký vẽ lên máy tính và lưu trữ lại…), một hình ảnh, một câu nói, hoặc một đoạn video và cũng có thể sử dụng hàm băm để lấy giá trị băm trước khi mã hóa.

Hiện nay có những nhà cung cấp dịch vụ tạo khóa mã và tạo chữ ký số cho những người cần sử dụng, họ chỉ cần trả phí. Tất nhiên chữ ký số có tính bảo mật thấp hơn vì nội dung cố định nên sau một thời gian có thể dùng phương pháp thống kê để thám mã. Để tăng độ bảo mật, người sử dụng có thể thường xuyên thay đổi nội dung chữ ký số. Chữ ký điện tử hay chữ ký số thường thuộc quyền sử dụng riêng của một người, giống như chữ ký thông thường.

Một tổ chức, một cơ quan hay doanh nghiệp cũng có thể tạo một chữ ký số sử dụng chung để xác nhận cho những thông điệp mà cơ quan mình phát hành. Nội dung chữ ký số dùng chung đó là logo biểu tượng của doanh nghiệp, một câu khẩu hiệu của tổ chức hoặc chính là con dấu của tổ chức đó. Vì vậy chữ ký số sử dụng chung cho tổ chức cũng được gọi là **con dấu số** của tổ chức.

Tuy nhiên phần nhiều trong các quy định pháp lý của giao dịch điện tử người ta không nói đến giá trị của con dấu số, nói khác đi, trong giao dịch điện tử không dùng con dấu số đi kèm với chữ ký số/chữ ký điện tử của người có trách nhiệm phát hành thông điệp của cơ quan tổ chức vì hai lý do sau đây:

- Chữ ký số/điện tử chỉ có một người biết và được quyền sử dụng trong khi con dấu số của một tổ chức (nếu có) thì rất nhiều người được quyền sử dụng, do vậy độ bảo mật của con dấu số thấp hơn chữ ký số.
- Trong một cơ quan, tổ chức, người giữ con dấu số thường có mức độ trách nhiệm thấp hơn nhiều so với những người dùng chữ ký số.

Vì vậy, nếu một tổ chức có tạo con dấu số sử dụng trong các thông điệp do cơ quan mình phát hành (kèm với chữ ký số của người có trách nhiệm trong tổ chức) thì cũng chỉ được xem như một sự xác nhận bổ sung không có giá trị tin cậy cao, giống như trong các văn bản thông thường của một tổ chức người ta dùng các giấy tờ, phong bì có in logo, tiêu đề của tổ chức vậy thôi!

5.2. HÀM BĂM

5.2.1. Khái niệm về hàm băm

Để lấy một bộ phận nhỏ của một thông điệp, ta sử dụng một phương pháp toán học gọi là phương pháp hàm băm (*Hash function*) là một giải thuật toán học (một ánh xạ một - một (một chiều)), cho ứng với mỗi khối dữ liệu (một dãy bit hay một đối tượng trong lập trình hướng đối tượng của thông điệp gốc) một giá trị băm duy nhất.

Chú ý ở đây tính một chiều có nghĩa là: Mỗi khối dữ liệu gốc qua một hàm băm sẽ cho một giá trị băm duy nhất, tuy vậy có thể có một giá trị băm ứng với hai khối dữ liệu gốc khác nhau vì vậy không thể từ giá trị băm tìm ngược lại khối dữ liệu đã sinh ra nó. Trường hợp qua một hàm băm H, nếu có hai khối dữ liệu gốc nào đó cho cùng một giá trị băm thì ta gọi đấy là một sự đụng độ.

Tuy nhiên điều quan trọng là: **Nếu hai giá trị băm khác nhau thì chắc chắn hai khối dữ liệu tạo ra chúng là khác nhau.** Vì vậy

người nhận có thể tính lại giá trị băm của thông điệp nhận được rồi so sánh với giá trị tính được khi giải mã chữ ký điện tử để kiểm tra: nếu hai giá trị khác nhau thì có thể khẳng định nội dung thông điệp đã bị thay đổi.

Một hàm băm được đánh giá là tốt nếu số đụng độ xảy ra rất nhỏ (xác suất rất thấp, hầu như không thể xảy ra).

Một vài kỹ thuật tính toán chẳng hạn như phân bố xác suất Poisson (phân bố xác suất tiệm cận cho các sự kiện hiếm hoi) có thể dùng để phân khả năng xảy ra đụng độ của những hàm băm khác nhau đối với những nhóm dữ liệu khác nhau. Về lý thuyết thì nói chung với mọi nhóm dữ liệu đều tồn tại một hàm băm được xem như là hàm băm "hoàn hảo" nhất cho nhóm dữ liệu đó. Một ***hàm băm hoàn hảo*** (theo định nghĩa) là hàm băm mà đối với mọi dữ liệu trong nhóm đang xét không tạo ra những giá trị băm trùng nhau. Nhưng trong thực tế rất khó để tìm được hàm băm hoàn hảo cho mỗi nhóm dữ liệu nên người ta thường bằng lòng những hàm băm "gần hoàn hảo" nghĩa là chỉ tạo ra một số rất ít đụng độ đối với từng nhóm dữ liệu (có thể kiểm tra được).

5.2.2. Các phương pháp tạo hàm băm

Một hàm băm tốt phải thỏa mãn các điều kiện sau:

- Tính toán nhanh
- Các khóa được phân bố đều trong bảng
- Ít xảy ra đụng độ
- Xử lý được các loại khóa có kiểu dữ liệu khác nhau.

Các hàm băm được xác định theo cách tạo ra giá trị băm từ một dữ liệu. Có hai phương pháp chính để tạo hàm băm thường dùng là phương pháp cộng và nhân và phương pháp quay vòng.

Phương pháp băm kiểu cộng và nhân

Theo phương pháp này giá trị băm được tạo ra bằng cách duyệt dọc theo chuỗi dữ liệu và liên tục cộng thêm vào một giá trị xuất phát từ một giá trị được tính cho mỗi phần tử trong dữ liệu. Giá trị tăng thêm ứng với mỗi phần tử thường được tính dưới dạng nhân với một số nguyên tố nào đó.

$$h(m) = h^{-1} \oplus (m \otimes p)$$

$$h(m) = \sum_{i=0}^{|m|} m_i \otimes p_i$$

$$h(m) = h^{-1} \otimes (m \otimes p)$$

$$h(m) = \prod_{i=0}^{|m|} m_i \otimes p_i$$

Phương pháp băm bằng cách quay vòng

Cũng cộng thêm vào mỗi phần tử trong dãy một giá trị giống như phương pháp băm kiểu cộng nhưng ở đây giá trị cộng thêm được xét từ cả hai phía bên trái và bên phải, tính toán để cộng thêm vào tại mỗi phần tử,

$$h(m) = h^{-1} \oplus (m << p) \otimes (m >> p)$$

$$h(m) = \sum_{i=0}^{|m|} (m_i << p_i) \otimes (m_i >> q_i)$$

$$h(m) = \prod_{i=0}^{|m|} (m_i << p_i) \otimes (m_i >> q_i)$$

Các dạng hàm băm thông dụng

Trong "Thư viện hàm băm tổng quát" (*The General Hash function Library*) có nêu lên một số hàm băm hỗn hợp cộng và quay vòng chẳng hạn như các thuật toán sau đây.

RS Hash Function: Một hàm băm đơn giản từ thuật toán Robert Sedgwicks.

JS Hash Function: Hàm băm tính từ hai phía do Justin Sobel đề xuất.

PJW Hash Function: Thuật toán băm dựa trên công trình của Peter J. Weinberger thuộc Phòng thí nghiệm AT&T Bell.

BKDR Hash Function: Hàm băm này được mô tả trong tác phẩm của Brian Kernighan và Dennis Ritchie's "The C Programming Language" (Ngôn ngữ lập trình C).

SDBM Hash Function: Đây là dạng hàm băm được chọn sử dụng trong các dự án mã nguồn mở SDBM.

DJB Hash Function do GS. Daniel J. Bernstein xây dựng và giới thiệu lần đầu tiên trên *newsgroup comp.lang.c*. Có lẽ đây là một trong những hàm băm hiệu quả nhất từ trước đến nay đã được công bố.

Message Digest (MD) algorithms: Những dãy thuật toán hướng byte, sản sinh ra một giá trị băm 128 bit cho các thông điệp có độ dài bất kỳ.

- *MD2 (RFC 1319):* được thiết kế cho những hệ thống có bộ nhớ hạn chế chẳng hạn như các thẻ thông minh.
- *MD4 (RFC 1320):* do Rivest phát triển, tương tự như MD2 nhưng được thiết kế đặc biệt cho những quá trình xử lý nhanh trong phần mềm.
- *MD5 (RFC 1321):* Cũng do Rivest phát triển sau khi phát hiện một số nhược điểm của MD4; sơ đồ này tương tự như MD4 nhưng hoạt động chậm hơn do phải xử lý nhiều trên dữ liệu gốc. MD5 được tích hợp vào nhiều sản phẩm dù rằng vẫn còn một số nhược điểm mà nhà mật mã học người Đức Hans Dobbertin đã chỉ ra năm 1996.

Secure Hash Algorithm (SHA): Thuật toán của chuẩn hàm băm an toàn của NIST. NIST's Secure Hash Standard (SHS). SHA-1 tạo ra một giá trị băm 160 bit ban đầu được công bố với tên gọi là FIPS 180-1 và RFC 3174. FIPS 180-2 (tức là SHA-2) mô tả 5 thuật toán trong chuẩn SHS: SHA-1 cùng với SHA-224, SHA-256, SHA-384, và SHA-512 có thể tạo ra giá trị băm có độ dài 224, 256, 384, hoặc 512 bit.

Chú ý: Năm 2004 một số nhà nghiên cứu đã phát hiện rằng đã có những sự đụng độ trong thực hành xảy ra đối với MD5, SHA-1, và một vài hàm băm khác!

RIPEMD: Một dãy thuật toán biến đổi thông điệp (*message digest*) thoạt đầu xuất phát từ dự án RIPE (*RACE Integrity Primitives Evaluation*). RIPEMD-160 do Hans Dobbertin, Antoon Bosselaers, và Bart Preneel thiết kế và tối ưu hóa cho quá trình xử lý 32 bit nhằm thay thế cho hàm băm 128 bit đang phổ biến thời đó. Có những phiên bản khác là RIPEMD-256, RIPEMD-320, và RIPEMD-128.

HAVAL (*HAsh of VAriable Length*): Hàm băm có độ dài biến thiên: Do Y. Zheng, J. Pieprzyk và J. Seberry, là một hàm băm với nhiều cấp độ an toàn khác nhau. HAVAL có thể tạo các giá trị băm với độ dài 128, 160, 192, 224, hoặc 256 bit.

Whirlpool: Là một hàm băm tương đối mới do V. Rijmen và P.S.L.M. Barreto thiết kế. Whirlpool làm việc trên các thông điệp có độ dài không quá 2^{256} bit và tạo ra giá trị băm với 512 bit. Thiết kế của Whirlpool rất khác biệt với thiết kế của MD5 và SHA-1, làm cho nó chống lại được những tấn công mà các hàm băm khác không chống được.

Tiger: Do Ross Anderson và Eli Biham thiết kế. Tiger được thiết kế đảm bảo an toàn cao chạy hiệu quả với bộ xử lý 64 bit nên đã thay thế dễ dàng MD4, MD5, SHA and SHA-1 trong những ứng dụng khác… Tiger/192 tạo nên đầu ra 192 bit và tương thích với kiến trúc

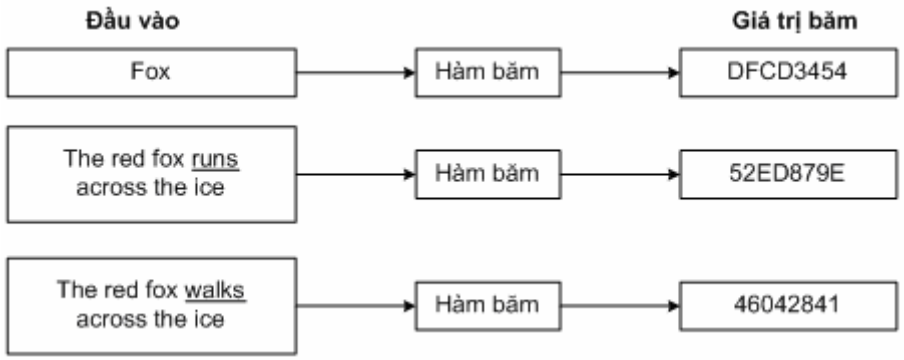
64 bit; Tiger/128 và Tiger/160 tạo ra giá trị băm có độ dài 128 và 160 bit, để tương thích với các hàm băm đã nêu trên.

5.2.3. Công dụng của hàm băm

Hàm băm thường được dùng để xây dựng các bảng băm tức là bảng ghi các giá trị băm ứng với một số khối dữ liệu mẫu: khi cần so sánh hai khối dữ liệu nào đó (thường có kích thước rất lớn) ta chỉ cần so sánh các giá trị băm có kích thước rất nhỏ của chúng: điều này rất có ích.

Ví dụ về hoạt động của một hàm băm:

Hai chuỗi dữ liệu gốc chỉ khác nhau một từ (*runs* và *walks* nhưng qua hàm băm cho ra hai giá trị băm hoàn toàn khác nhau. So sánh hai giá trị băm thấy khác nhau ta biết ngay hai chuỗi dữ liệu gốc là khác nhau (dù không thể biết chúng khác nhau ở đâu!)



Vì tính thông dụng của bảng băm, ngày nay, đa số ngôn ngữ lập trình đều cung cấp thư viện ứng dụng bảng băm, trong đó có các vấn đề như: bộ sưu tập (*collection*), các danh sách (*list*), các bảng (*table*), các ánh xạ (*mapping*), các từ điển (*dictionary*).

Thông thường, các lập trình viên chỉ cần viết hàm băm cho các đối tượng nhằm tích hợp với thư viện bảng băm đã được xây dựng

sẵn. Bảng băm là một ứng dụng quan trọng của các hàm băm, cho phép tra cứu nhanh một bản ghi dữ liệu nếu cho trước *khóa mã* của bản ghi đó (Lưu ý: các khóa này thường không bí mật và được dùng để "mở khóa" hoặc để truy nhập thông tin).

Các hàm băm dành cho việc phát hiện và sửa lỗi tập trung phân biệt các trường hợp mà dữ liệu đã bị làm nhiễu trong quá trình truyền tin, giá trị băm tương đối nhỏ có thể được dùng để kiểm chứng rằng một tệp dữ liệu có kích thước tùy ý đã bị sửa đổi hay không. Hàm băm được dùng để phát hiện lỗi truyền dữ liệu như sau. Phía bên gửi, hàm băm được tính cho dữ liệu được gửi, giá trị băm này được gửi cùng dữ liệu. Phía bên nhận, hàm băm lại được tính lần nữa, nếu các giá trị băm không trùng nhau thì lỗi đã xảy ra ở đâu đó trong quá trình truyền.

Việc này gọi là **kiểm tra thặng dư** (*redundancy check*).

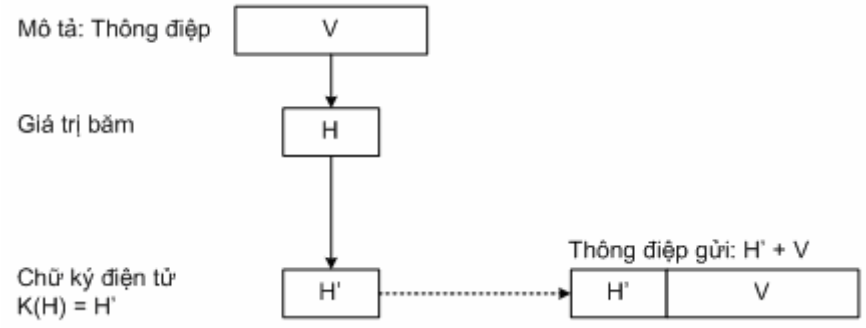
- Giả sử An có thông điệp V cần gửi cho Bình. An lấy giá trị băm H(V), mã hóa bằng khóa K đã trao đổi với Bình: K[H(V)] = H' gắn với V và gửi tất cả cho Bình. H' chính là chữ ký điện tử của An trong thông điệp.

- Bình nhận được thông điệp, trước hết giải mã H' tìm ra một giá trị băm H. Tiếp đó dùng hàm băm chung tính lại giá trị băm của thông điệp nhận được. Nếu giá trị băm đó trùng với giá trị H nói trên thì Bình có hai kết luận:

1. Thông điệp chính do An gửi (qua kiểm tra khóa K).
2. Thông điệp không bị thay đổi trong quá trình truyền tin (giá trị băm trùng nhau). Trường hợp hai giá trị tìm được khác nhau thì:

- Hoặc không phải là thông điệp do An gửi.

- Nội dung thông điệp đã sai lạc trong quá trình truyền.



Hình 5.2: Sơ đồ tạo chữ ký điện tử

Sử dụng chữ ký điện tử (có chứng thực) gắn kèm vào thông điệp, có thể đảm bảo các yêu cầu:

- Nhận diện định danh người phát hành thông điệp

- Người phát hành không thể chối bỏ

- Đảm bảo tính toàn vẹn thông tin, phát hiện được trường hợp thông điệp bị can thiệp trên đường chuyển vận.

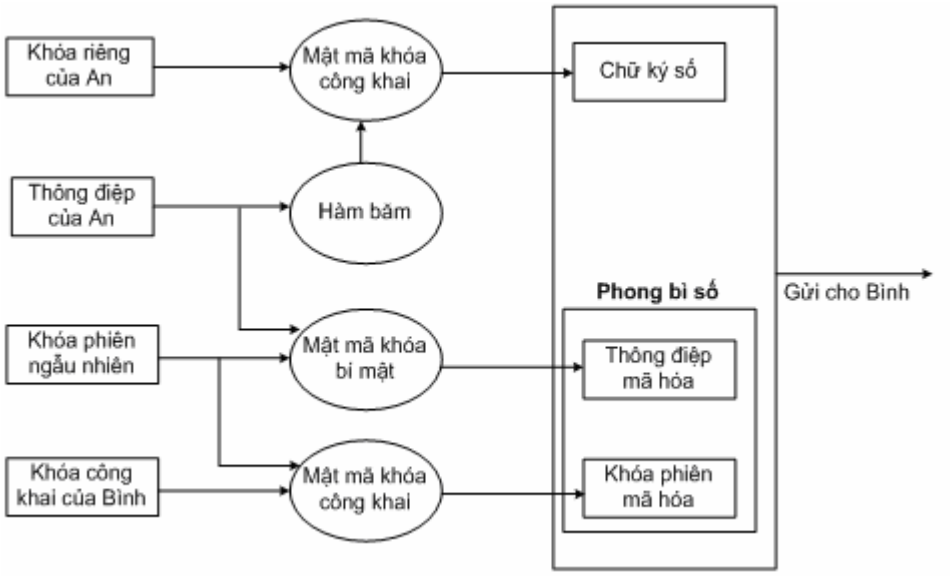
5.2.4. Phong bì số an toàn

Ở mục 4.2.4 ta đã thấy rằng có thể dùng sơ đồ trao đổi khóa công khai giữa hai đối tác để tạo một phong bì số khá đơn giản dùng để chuyển giao khóa đối xứng. Tuy nhiên dạng phong bì số đó không an toàn vì có khả năng bị tấn công của người đứng giữa.

Bằng cách phối hợp cả hai loại khóa mã hóa đối xứng, bất đối xứng với thuật toán hàm băm, ta có thể tạo được một sơ đồ giao dịch điện tử an toàn đảm bảo được các yêu cầu của các nguyên lý bảo mật trong giao dịch.

Hình 5.3 mô tả một quá trình mã hóa hỗn hợp giữa 3 loại thuật toán mã hóa đối xứng, bất đối xứng và hàm băm để tạo ra một chữ

ký điện tử và một phong bì số. Trong ví dụ ở hình 5.3, An là người gửi còn Bình là người nhận.



Hình 5.3: Phối hợp hàm băm và hai loại mã hóa đối xứng, bất đối xứng tạo phong bì số an toàn trong giao dịch điện tử

Một phong bì số bao gồm một thông điệp mã hóa và một khóa phiên được mã hóa. An dùng khóa mã hóa bí mật để mã hóa thông điệp đã mã hóa sử dụng khóa phiên mà An tạo ra một cách ngẫu nhiên cho từng phiên giao dịch. An dùng khóa công khai của Bình để mã hóa khóa phiên. Cả thông điệp mã hóa và cả khóa phiên tạo nên phong bì số. Khi Bình nhận thông điệp, Bình sẽ dùng khóa riêng của mình để giải mã.

Chữ ký điện tử được tạo thành bởi 2 bước. Trước tiên An tính toán giá trị băm của thông điệp, tiếp đó An mã hóa giá trị băm đó bằng khóa bí mật của mình.

Khi nhận được chữ ký số, Bình dùng khóa mã đã thỏa thuận với An để giải mã chữ ký số và tìm lại được giá trị băm thông điệp của An. Bình lại dùng khóa của An giải mã tìm khóa phiên và tiếp đó

giải mã thông điệp của An sử dụng khóa phiên và khóa mã đã trao đổi với An.

Sau cùng Bình tìm lại giá trị băm của thông điệp đã giải mã và so sánh với giá trị băm có được từ việc giải mã chữ ký điện tử. Nếu hai giá trị là trùng nhau thì Bình tin chắc là thông điệp toàn vẹn, không bị can thiệp trong quá trình truyền, mặt khác cũng đảm bảo thông điệp đúng là của An.

Về phần An, sau khi gửi thông điệp hoàn toàn tin tưởng rằng chắc chắn chỉ có Bình mới giải mã toàn bộ và đọc được thông điệp của mình vì chỉ có Bình có khóa bí mật.

5.3. HẠ TẦNG CƠ SỞ KHÓA CÔNG KHAI

5.3.1. Nhu cầu chứng thực trong giao dịch điện tử

Như đã trình bày ở chương 3, vấn đề phân phối, trao đổi khóa mã (thường là khóa đối xứng) trước khi tiến hành giao dịch điện tử an toàn là điều rất khó khăn và thường xuyên đối mặt với nhiều hiểm họa.

Giả sử An và Bình trao đổi khóa (công khai) và chữ ký điện tử để giao dịch với nhau. Việc giao dịch tiến hành bình thường: mỗi bên đối tác hoàn toàn nhận biết các thông điệp do phía bên đối tác kia phát hành.

Tuy nhiên việc trao đổi khóa công khai và chữ ký điện tử là giao dịch hoàn toàn riêng tư giữa 2 đối tác nên có nguy cơ là nếu một trong hai phía chối bỏ những thông điệp do mình phát hành bằng cách không thừa nhận khóa công khai và chữ ký điện tử đã trao đổi là của mình thì việc giao dịch đổ vỡ và không thể quy trách nhiệm pháp lý cho ai được.

Phát sinh nhu cầu là phải có bên thứ ba (trent) "làm chứng" cho việc trao đổi đó. Nói một cách khác, khi một cá nhân, pháp nhân muốn sử dụng khóa mã và chữ ký điện tử của mình để xác định trách

nhiệm đối với các thông điệp do mình phát hành thì phải có sự chứng thực của một tổ chức có trách nhiệm và quyền lực nào đó.

Cơ quan chứng thực điện tử CA (*Certification Authority*) là một tổ chức đóng vai trò "bên thứ ba" trong các giao dịch điện tử. Muốn được tín nhiệm trong giao dịch, một tổ chức/cá nhân phải đến đăng ký với một CA.

CA cấp các chứng thư số (còn gọi là **chứng thực điện tử - chứng thư**) xác nhận việc sử dụng chữ ký số và gắn một khóa công khai với một thực thể (cá nhân, pháp nhân, hoặc máy chủ cung cấp dịch vụ...)

Một chứng thực khóa công khai tiêu biểu thường bao gồm khóa công khai và các thông tin (tên, địa chỉ...) về thực thể sở hữu khóa đó.

Chứng thư điện tử còn có thể được sử dụng để kiểm tra một khóa công khai nào đó thuộc về ai.

Trong một mô hình hạ tầng khóa công khai (PKI) tiêu biểu, chữ ký trong chứng thực thuộc về cơ quan cấp chứng thực số CA thì Trent chính là cơ quan cấp chứng thực số.

Trong mô hình mạng lưới tín nhiệm (Web of trust), thì chữ ký có thể là của chính thực thể đó hoặc của một thực thể khác, Trent có thể là bất kỳ người dùng nào và mức độ tin tưởng tùy thuộc vào sự đánh giá của người dùng.

Trong bất kỳ trường hợp nào thì chữ ký trong chứng thực là sự đảm bảo của người ký về mối liên hệ giữa khóa công khai và thực thể được chứng nhận.

Việc sử dụng chứng thực sẽ tạo điều kiện áp dụng rộng rãi mật mã hóa khóa công khai. Đối với hệ thống mã hóa khóa bí mật (khóa đối xứng), việc trao đổi khóa giữa những người sử dụng trên quy mô lớn là không thể thực hiện được. Hệ thống mã hóa khóa công khai có thể tránh được vấn đề này.

Như trong ví dụ xét ở trên, về nguyên tắc nếu An và Bình muốn người khác gửi thông tin mật cho mình thì chỉ cần công bố khóa công khai của chính mình. Bất kỳ ai có được khóa này đều có thể gửi thông tin mật cho họ. Tuy nhiên, bất kỳ người nào khác (Công chẳng hạn) cũng có khả năng đưa ra một khóa công khai khác và giả mạo rằng đó là khóa của An. Bằng cách làm như vậy kẻ tấn công có thể đọc được một số thông tin gửi cho An. Nhưng nếu An đưa khóa công khai của mình vào một chứng thực và chứng thực này được một bên thứ ba (*Trent*) xác nhận bằng chữ ký điện tử thì bất kỳ ai tin tưởng vào Trent sẽ có thể kiểm tra xem khóa công khai đó có đúng là của An không.

5.3.2. Hạ tầng cơ sở khóa công khai

Trong mật mã học, hạ tầng cơ sở khóa công khai PKI (Public Key Infrastructure) là một cơ chế để cho một bên thứ 3 (thường là cơ quan cấp chứng thực số) cung cấp và xác thực định danh các bên tham gia vào quá trình trao đổi thông tin. Cơ chế này cũng cho phép gán cho mỗi người sử dụng trong hệ thống một cặp khóa công khai/khóa bí mật. Các quá trình này thường được thực hiện bởi một phần mềm đặt tại trung tâm và các phần mềm phối hợp khác tại các địa điểm của người dùng.

Khóa công khai thường được phân phối trong chứng thực điện tử. Khái niệm hạ tầng khóa công khai thường được dùng để chỉ toàn bộ hệ thống bao gồm cơ quan cấp chứng thực số (CA) cùng các cơ chế liên quan đồng thời với toàn bộ việc sử dụng các thuật toán mật mã hóa khóa công khai trong trao đổi thông tin. Tuy nhiên phần sau được bao gồm không hoàn toàn chính xác bởi vì các cơ chế trong PKI không nhất thiết sử dụng các thuật toán mã hóa khóa công khai. PKI cho phép những người tham gia xác thực lẫn nhau và sử dụng thông tin từ các chứng thực khóa công khai để mã hóa và giải mã thông tin trong quá trình trao đổi. Thông thường, PKI bao gồm phần mềm máy

khách (client), phần mềm máy chủ (server), phần cứng (như thẻ thông minh) và các quy trình hoạt động liên quan. Người sử dụng cũng có thể ký các văn bản điện tử với khóa bí mật của mình và mọi người đều có thể kiểm tra với khóa công khai của họ.

PKI cho phép các giao dịch điện tử được diễn ra đảm bảo tính bí mật, toàn vẹn và xác thực lẫn nhau mà không cần phải trao đổi các thông tin mật từ trước. Hầu hết các hệ thống PKI quy mô doanh nghiệp đều dựa trên các chuỗi chứng thực để xác thực các thực thể. Chứng thực của người dùng sẽ được một cơ quan cấp chứng thực số cấp, đến lượt nhà cung cấp này lại có chứng thực được một nhà cung cấp khác ở cấp cao hơn tạo ra... (hình cây). Hệ thống sẽ bao gồm nhiều máy tính thuộc nhiều tổ chức khác nhau với các gói phần mềm tương thích từ nhiều nguồn khác nhau. Vì vậy, các tiêu chuẩn là yếu tố rất quan trọng đối với hoạt động của các PKI. Hầu hết các tiêu chuẩn về PKI hiện tại được soạn thảo bởi nhóm làm việc PKIX của IETF.

Các hệ thống PKI doanh nghiệp thường được tổ chức theo mô hình danh bạ trong đó khóa công khai của mỗi người dùng được lưu trữ (bên trong các chứng thực số) kèm với các thông tin cá nhân (số điện thoại, E-mail, địa chỉ, nơi làm việc...). Hiện nay, công nghệ danh bạ tiên tiến nhất là LDAP và định dạng chứng thực phổ biến nhất X.509 cũng được phát triển từ mô hình trước đó của LDAP là X.500.

Mục tiêu chính của PKI là cung cấp khóa công khai và xác định mối liên hệ giữa khóa và định dạng người dùng. Nhờ vậy người dùng có thể sử dụng trong một số ứng dụng như:

- Mã hóa E-mail hoặc xác thực người gửi E-mail (OpenPGP hay S/MIME).
- Mã hóa hoặc nhận thực văn bản (Các tiêu chuẩn chữ ký XML hoặc mã hóa XML khi văn bản được thể hiện dưới dạng XML).

- Xác thực người dùng ứng dụng (Đăng nhập bằng thẻ thông minh nhận thực người dùng trong SSL).
- Các giao thức truyền thông an toàn dùng kỹ thuật Bootstrapping (IKE, SSL): trao đổi khóa bằng khóa bất đối xứng, còn mã hóa bằng khóa đối xứng.

5.3.3. Chuẩn X.509

Khi áp dụng chứng thực ở quy mô lớn, có rất nhiều CA cùng hoạt động. Vì vậy chẳng hạn như An có thể không quen thuộc (không đủ tin tưởng) với CA của Bình. Do đó chứng thực của Bình có thể phải bao gồm chữ ký của CA ở mức cao hơn. Quá trình này dẫn đến việc hình thành một hệ thống mạng lưới quan hệ phức tạp và phân tầng giữa các CA. Một hệ thống tổ chức như vậy là một cơ sở hạ tầng kiến trúc khóa công khai PKI. PKI là một kiến trúc phân cấp những đối tượng có trách nhiệm xác minh các khóa công khai lẫn nhau.

Chuẩn chứng thực khóa công khai phổ biến nhất hiện nay là X.509 do *ITU-T* ban hành. Chuẩn này được làm thích ứng với Internet bởi nhóm công tác *IETF PKIX working group*. X.509 là một đề nghị của ITU Liên minh Viễn thông Quốc tế (ITU) định nghĩa một hoạt động khung (*framework*) về chứng thực. Thực ra hiện tại chuẩn X.509 đang được diễn giải theo một số cách, tùy theo công ty cung cấp quyết định sử dụng như thế nào.

X.509 lần đầu tiên được công bố vào năm 1988, và các phiên bản tiếp theo đã được đưa ra để giải quyết các vấn đề an toàn. X.509 hỗ trợ cả hai mã bí mật (mã đơn) và mã công khai. X.509 định nghĩa các nội dung về một chứng thực, bao gồm số phiên bản, số serial, ID chữ ký, tên công bố, thời điểm có hiệu lực, định nghĩa chủ đề, phần mở rộng và chữ ký trên các trường trên.

Về cơ bản, một người có trách nhiệm chứng nhận sẽ đặt khóa công khai của một người nào đó có nhu cầu chứng thực vào thủ tục chứng thực, sau đó xác thực lại bằng khóa riêng. Điều này bắt buộc khóa và thủ tục chứng thực phải luôn đi kèm với nhau. Bất cứ ai cần dùng khóa công cộng của một đối tượng nào đó đều có thể mở thủ tục chứng thực bằng khóa công cộng của các đối tượng này do người có trách nhiệm chứng thực cung cấp (các khóa công cộng này được ký hoặc khóa bằng khóa riêng của người có trách nhiệm chứng thực).

Vì vậy, người sử dụng phải tin rằng người có trách nhiệm chứng thực sẽ bảo đảm việc hợp lệ hóa người chủ của khóa công khai và thực sự khóa công khai ở đây chính là khóa công khai của người có trách nhiệm chứng thực. Đây chính là lãnh địa của các PKI.

Trong chuẩn X.509 về hệ thống hạ tầng khóa công khai, mạng lưới CA tạo thành cây từ trên xuống với gốc là một CA trung tâm đầu tiên *CA gốc* (*Root CA*), không cần được chứng thực bởi một bên nào khác.

5.3.4. Thu hồi khóa

Một chứng thực khóa công khai có thể bị thu hồi nếu như khóa riêng tương ứng của nó đã bị lộ hoặc mối liên hệ giữa khóa công khai và chủ thể sở hữu đã thay đổi. Điều này có thể xảy ra ở mức độ không thường xuyên nhưng người sử dụng phải luôn kiểm tra tính pháp lý của chứng thực mỗi khi sử dụng.

Việc kiểm tra này có thể thực hiện bằng cách so sánh chứng thực cụ thể cần xem xét với danh sách các chứng thực bị thu hồi *CRL* (*Certificate Revocation List*). Việc đảm bảo danh sách này chính xác và cập nhật là chức năng cơ bản của hạ tầng khóa công cộng tập trung. Tuy nhiên công việc này đòi hỏi tốn kém lớn về nhân công cũng như ngân sách nên thường không được thực hiện đầy đủ. Để

thực sự đạt hiệu quả, danh sách này phải luôn sẵn sàng cho bất kỳ ai cần đến vào bất kỳ thời điểm nào tại mọi nơi.

Một cách kiểm tra khác là truy vấn vào nơi đã cung cấp chứng thực với giao thức kiểm tra chứng thực trực tuyến *OCSP* (*Online Certificate Status Protocol*).

Cả hai phương pháp trên đều có thể bị thay thế bằng một chuẩn mới là chuẩn XKMS. Tuy nhiên tiêu chuẩn XKMS này hiện nay còn chưa được sử dụng rộng rãi. Một chứng thực số tiêu biểu gồm các thành phần sau:

- Khóa công khai;
- Tên: có thể là tên người, máy chủ hoặc tổ chức;
- Thời hạn sử dụng;
- Địa chỉ URL của trung tâm thu hồi chứng thực (để kiểm tra).

5.3.5. Cơ quan cấp chứng thực số tự động

Các rô bốt CA (*Robot CA*) là các chương trình máy tính tự động có khả năng kiểm tra và xác nhận một số khía cạnh của khóa công cộng. Các rô bốt này có thể làm giảm đáng kể những tấn công vào hệ thống, đặc biệt là các tấn công nhằm vào việc làm chệch hướng các luồng thông tin trên mạng. Các khía cạnh của khóa công cộng thường được kiểm tra là:

- Khóa được công bố dưới nhận thức của người sở hữu địa chỉ E-mail gắn với khóa,
- Người sở hữu địa chỉ E-mail đang có khóa bí mật,
- Tình trạng sử dụng khóa.

Phân loại chứng thực số

Công ty Verisign đưa ra mô hình gồm 3 loại chứng thực điện tử sau đây:

- Loại 1 dành cho cá nhân, dự kiến dùng gắn vào cho E-mail.
- Loại 2 dành cho tổ chức với yêu cầu chứng minh nguồn gốc và tư cách pháp nhân.
- Loại 3 dành cho máy chủ và phần mềm với khả năng kiểm tra độc lập bằng cách truy vấn tới CA nơi cung cấp.

5.3.6. Một số hệ thống PKI

Việc *Diffie*, *Hellman* và *Rivest*, *Shamir*, *Adleman* công bố công trình nghiên cứu về *trao đổi khóa* an toàn và thuật toán mật mã hóa khóa công khai vào năm 1976 (Chương 3) đã làm thay đổi hoàn toàn cách thức trao đổi thông tin mật. Cùng với sự phát triển của các hệ thống truyền thông điện tử tốc độ cao (Internet và các hệ thống trước nó), nhu cầu về trao đổi thông tin bí mật trở nên cấp thiết. Thêm vào đó một yêu cầu nữa phát sinh là việc xác định định dạng của những người tham gia vào quá trình thông tin. Vì vậy ý tưởng về việc gắn định dạng người dùng với chứng thực được bảo vệ bằng các kỹ thuật mật mã đã được phát triển một cách mạnh mẽ.

Các nhà doanh nghiệp kỳ vọng vào một thị trường hứa hẹn mới đã thành lập những công ty hoặc dự án mới về PKI và bắt đầu vận động các chính phủ để hình thành nên khung pháp lý về lĩnh vực này.

Một dự án của ***American Bar Association*** đã xuất bản một nghiên cứu tổng quát về những vấn đề pháp lý có thể nảy sinh khi vận hành PKI. Không lâu sau đó, một vài tiểu bang của Hoa Kỳ mà đi đầu là Utah (năm 1995) đã thông qua những dự luật và quy định đầu tiên.

Các nhóm bảo vệ quyền lợi người tiêu dùng thì đặt ra các vấn đề về bảo vệ quyền riêng tư và các trách nhiệm pháp lý. Tuy nhiên, các luật và quy định đã được thông qua lại không thống nhất trên thế giới. Thêm vào đó là những khó khăn về kỹ thuật và vận hành khiến cho việc thực hiện PKI khó khăn hơn rất nhiều so với kỳ vọng ban đầu.

Tại thời điểm đầu thế kỷ XXI, người ta nhận ra rằng các kỹ thuật mật mã cũng như các quy trình/giao thức rất khó được thực hiện chính xác và các tiêu chuẩn hiện tại chưa đáp ứng được các yêu cầu đề ra. Thị trường PKI thực sự đã tồn tại và phát triển nhưng không phải với quy mô đã được kỳ vọng từ những năm giữa của thập kỷ 1990. PKI chưa giải quyết được một số vấn đề mà người ta đã đặt hy vọng vào nó.

Những PKI thành công nhất tới nay là các phiên bản do các chính phủ thực hiện.

Dưới đây là danh sách một số hệ thống PKI, trong đó một số cơ quan cấp chứng thực số hàng đầu (ví dụ VeriSign) không được liệt kê vì các phần mềm của họ không được công bố công khai.

- Hệ thống quản lý chứng thực Red Hat
- Computer Associates eTrust PKI
- Entrust
- Microsoft
- US Government External Certificate Authority (ECA)
- Nexus
- OpenCA (Một mô hình PKI mã nguồn mở)
- RSA Security
- phpki
- GenCerti
- ejbca
- newpki
- Papyrus CA Software
- pyCA
- IDX-PKI

- TinyCA
- ElyCA
- SimpleCA
- SeguriData
- Safelayer Secure Communications

5.4. GIAO THỨC PGP VÀ MẠNG LƯỚI TIN CẬY

5.4.1. Chuẩn PGP

Chuẩn PGP (Pretty Goods Privacy) là một chương trình máy tính mã hóa và giải mã các dữ liệu được truyền trên các E-mail cần bảo mật, do Phillip Zimmermann đề xuất năm 1991 và là một trong những chương trình đang được phát triển rộng rãi và hiện nay Tập đoàn PGP đang cung cấp nhiều phần mềm dựa trên nền tảng này.

Với mục tiêu ban đầu là phục vụ cho mã hóa thư điện tử, PGP đến nay đã trở thành một giải pháp mã hóa cho các chính phủ, các công ty lớn cũng như các cá nhân. Các phần mềm dựa trên PGP được dùng để mã hóa và bảo vệ thông tin lưu trữ trên máy tính xách tay, máy tính để bàn, máy chủ và trong quá trình trao đổi thông qua E-mail, YM hoặc chuyển file. Giao thức hoạt động của hệ thống này có ảnh hưởng lớn và trở thành một trong hai tiêu chuẩn mã hóa thư điện tử (tiêu chuẩn còn lại là S/MIME, được sử dụng phổ biến hơn).

Mã hóa PGP sử dụng một tổ hợp các thuật toán hàm băm, nén dữ liệu, mã hóa khóa đối xứng và cuối cùng là tạo ra các cặp khóa riêng và khóa công khai cộng thêm với hệ thống xác lập mối quan hệ giữa khóa công khai và chỉ danh người dùng (“căn cước” - ID). Phiên bản đầu tiên của hệ thống này thường được biết dưới tên mạng lưới tin cậy dựa trên các mối quan hệ ngang hàng (khác với hệ thống X.509 với cấu trúc cây dựa vào các cơ quan cấp chứng thực số). Các

phiên bản PGP về sau dựa trên các kiến trúc tương tự như hạ tầng khóa công khai.

Ban đầu PGP nhằm vào mục tiêu chủ yếu là mật mã hóa nội dung các thông điệp thư điện tử và các tệp đính kèm cho người dùng phổ thông. Bắt đầu từ 2002, các sản phẩm PGP đã được đa dạng hóa thành một tập hợp ứng dụng mật mã và có thể được đặt dưới sự quản trị của một máy chủ. Các ứng dụng PGP giờ đây bao gồm: thư điện tử, chữ ký số, mật mã hóa ổ đĩa cứng máy tính xách tay, bảo mật tệp và thư mục, bảo mật các phiên trao đổi YM, mật mã hóa luồng chuyển tệp, bảo vệ các tệp và thư mục lưu trữ trên máy chủ mạng.

Phiên bản PGP Desktop 9.x dành cho máy để bàn bao gồm các tính năng: thư điện tử, chữ ký số, bảo mật YM, mật mã hóa ổ đĩa cứng máy tính xách tay, bảo mật tệp và thư mục, tệp nén tự giải mã, xóa file an toàn. Các tính năng riêng biệt được cấp phép theo các cách khác nhau tùy theo yêu cầu.

Phiên bản PGP Universal 2.x dành cho máy chủ cho phép triển khai ứng dụng tập trung, thiết lập chính sách an ninh và lập báo cáo. Phần mềm này được dùng để mật mã hóa thư điện tử một cách tự động tại cổng ra vào (*gateway*) và quản lý các phần mềm máy khách PGP Desktop 9.x. Nó làm việc với máy chủ khóa công khai PGP (gọi là *PGP Global Directory*) để tìm kiếm khóa của người nhận và có khả năng gửi thư điện tử an toàn ngay cả khi không tìm thấy khóa của người nhận bằng cách sử dụng phiên làm việc HTTPS.

Với ứng dụng PGP Desktop 9.0 được quản lý bởi PGP Universal Server 2.0, tất cả các ứng dụng mật mã hóa PGP được dựa trên nền kiến trúc proxy mới. Các phần mềm này giúp loại bỏ việc sử dụng các plug-in của thư điện tử và tránh cho người dùng việc sử dụng các ứng dụng khác. Tất cả các hoạt động của máy chủ cũng như máy khách đều tự động tuân theo một chính sách an ninh. PGP Universal server còn tự động hóa các quá trình tạo, quản lý và kết thúc các khóa chia sẻ giữa các ứng dụng PGP.

Các phiên bản mới của PGP cho phép sử dụng cả 2 tiêu chuẩn: OpenPGP và S/MIME, cho phép trao đổi với bất kỳ ứng dụng nào tuân theo tiêu chuẩn của NIST.

5.4.2. Hoạt động của PGP

Như đã nói trên, PGP sử dụng kết hợp mật mã hóa khóa công khai và thuật toán khóa đối xứng. Trong các hệ thống này, người sử dụng đầu tiên phải có một cặp khóa: khóa công khai và khóa bí mật. Người gửi sử dụng khóa công khai của người nhận để mã hóa một khóa chung (khóa phiên - *session key*) dùng trong các thuật toán mật mã hóa khóa đối xứng. Mỗi khóa công khai do PGP tạo ra chỉ được trao cho một người dùng (*user)* hay một địa chỉ E-mail. Phiên bản đầu tiên của hệ thống này có tên gọi là mạng lưới tin cậy được đảm bảo bởi các khóa công khai cấp cho mọi thành viên khi đăng ký gia nhập. Mỗi người sử dụng đều có thể đăng ký một mật khẩu riêng tùy ý để sử dụng khóa công khai đã được cấp nhưng nếu một người sử dụng biết mật khẩu riêng của người sử dụng khác thì họ vẫn có thể dùng khóa công khai đã được cấp cho người đó. Vì vậy hệ thống này có nhược điểm là chỉ đảm bảo an toàn với số người sử dụng không quá lớn để có thể kiểm soát được việc sử dụng mật khẩu cá nhân bằng những biện pháp đi kèm khác.

PGP gửi những thông điệp bí mật được mã hóa bằng một khóa đối xứng, khóa này chỉ sử dụng một lần gọi là khóa phiên (*session key*). Khóa phiên được mã hóa bằng khóa riêng tương ứng với khóa công khai đã được phân phối trước cho người dùng (hoặc cho địa chỉ E-mail cần gửi). Người dùng sử dụng khóa công khai mình được cấp, giải mã để tìm khóa phiên và tiếp đó dùng khóa phiên để giải mã thông điệp.

Khóa phiên này chính là khóa để mật mã hóa các thông tin được gửi qua lại trong phiên giao dịch. Rất nhiều khóa công khai của

những người sử dụng PGP được lưu trữ trên các máy chủ khóa PGP trên khắp thế giới (các máy chủ "soi" - *mirror* lẫn nhau).

Người nhận trong hệ thống PGP sử dụng khóa phiên để giải mã các gói tin. Khóa phiên này cũng được gửi kèm với thông điệp nhưng được mật mã hóa bằng hệ thống mật mã bất đối xứng và có thể tự giải mã với khóa bí mật của người nhận. Hệ thống phải sử dụng cả 2 dạng thuật toán để tận dụng ưu thế của cả hai: thuật toán bất đối xứng đơn giản việc phân phối khóa còn thuật toán đối xứng có ưu thế về tốc độ (nhanh hơn cỡ 1000 lần).

Một chiến lược tương tự cũng được dùng (mặc định) để phát hiện xem thông điệp có bị thay đổi hoặc giả mạo người gửi. Để thực hiện 2 mục tiêu trên người gửi phải ký văn bản với thuật toán RSA hoặc DSA. Đầu tiên, PGP tính giá trị hàm băm của thông điệp rồi tạo ra chữ ký số với khóa bí mật của người gửi. Khi nhận được văn bản, người nhận tính lại giá trị băm của văn bản đó đồng thời giải mã chữ ký số bằng khóa công khai của người gửi. Nếu 2 giá trị này giống nhau thì có thể khẳng định (với xác suất rất cao) là văn bản chưa bị thay đổi kể từ khi gửi và người gửi đúng là người sở hữu khóa bí mật tương ứng.

Trong quá trình mã hóa cũng như kiểm tra chữ ký, một điều vô cùng quan trọng là khóa công khai được sử dụng thực sự thuộc về người được cho là sở hữu nó. Nếu chỉ đơn giản là tải về (*download*) một khóa công khai từ đâu đó sẽ không thể đảm bảo được điều này. PGP thực hiện việc phân phối khóa thông qua chứng thực số được tạo nên bởi những kỹ thuật mật mã sao cho việc sửa đổi (không hợp pháp) có thể dễ dàng bị phát hiện. Tuy nhiên chỉ điều này thôi thì chưa đủ vì nó chỉ ngăn chặn được việc sửa đổi sau khi chứng thực đã được tạo ra. Người dùng còn cần phải được trang bị khả năng kiểm tra xem khóa công khai có thực sự thuộc về người được cho là sở hữu hay không. Từ phiên bản đầu tiên, PGP đã có một cơ chế hỗ trợ

điều này là mạng lưới tin cậy. Mỗi khóa công khai (rộng hơn là các thông tin gắn với một khóa hay một người) đều có thể được một bên thứ 3 xác nhận (điện tử).

Trong các đặc tả gần đây của OpenPGP, các chữ ký tin cậy có thể được sử dụng để tạo ra do các cơ quan cấp chứng thực số (CA). Một chữ ký tin cậy có thể chứng tỏ rằng một khóa thực sự thuộc về một người sử dụng và người đó đáng tin cậy để ký xác nhận một khóa của mức thấp hơn. Một chữ ký có mức 0 tương đương với chữ ký trong mô hình mạng lưới tín nhiệm. Chữ ký ở mức 1 tương đương với chữ ký của một CA vì nó có khả năng xác nhận cho một số lượng không hạn chế chữ ký ở mức 0. Chữ ký ở mức 2 tương tự như chữ ký trong danh sách các CA mặc định trong Internet Explorer; nó cho phép người chủ tạo ra các CA khác.

PGP cũng được thiết kế với khả năng hủy bỏ/thu hồi các chứng thực có khả năng đã bị vô hiệu hóa. Về một khía cạnh nào đó, điều này tương đương với danh sách chứng thực bị thu hồi của mô hình hạ tầng khóa công khai. Các phiên bản PGP gần đây cũng hỗ trợ tính năng hạn sử dụng của chứng thực.

Vấn đề xác định mối quan hệ giữa khóa công khai và người sở hữu không phải là vấn đề riêng của PGP. Tất cả các hệ thống sử dụng khóa công khai/bí mật đều phải đối phó với vấn đề này và cho đến nay chưa có một giải pháp hoàn thiện nào được tìm ra. Mô hình ban đầu của PGP trao cho quyền quyết định cuối cùng người sử dụng còn các mô hình PKI thì quy định tất cả các chứng thực phải được xác nhận (có thể không trực tiếp) bởi một nhà cung cấp chứng thực trung tâm.

5.4.3. An toàn bảo mật

Khi được sử dụng đúng quy cách, PGP được xem là có độ an toàn rất cao. Hiện nay chưa có phương pháp nào được biết tới có khả năng phá vỡ được PGP ở tất cả các phiên bản. Năm 1996, nhà mật

mã học Bruce Schneier đánh giá các phiên bản đầu tiên của PGP là "thứ gần nhất với mật mã hóa của quân đội mà mọi người có được" (*Applied Cryptography*, xuất bản lần 2, trang 587).

Trái với những hệ thống an ninh/giao thức như SSL chỉ nhằm bảo vệ thông tin trên đường truyền, PGP có thể bảo vệ cả dữ liệu cho mục đích lưu trữ lâu dài (hệ thống file).

Cũng giống như các hệ thống mật mã và phần mềm khác, an ninh của PGP có thể bị vô hiệu trong trường hợp sử dụng sai hoặc thông qua các dạng tấn công gián tiếp. Trong một trường hợp, FBI đã được tòa án cho phép cài đặt bí mật phần mềm ghi nhận bàn phím (*keystroke logging*) để thu thập mật khẩu PGP của người bị tình nghi. Sau đó, toàn bộ các tệp/E-mail của người đó bị vô hiệu và là chứng cứ pháp lý để định tội danh.

Ngoài những vấn đề trên, về khía cạnh mật mã học, an ninh của PGP phụ thuộc vào các giả định về thuật toán mà nó sử dụng trong điều kiện về thiết bị và kỹ thuật đương thời. Chẳng hạn, phiên bản PGP đầu tiên sử dụng thuật toán RSA để mã hóa khóa phiên; an ninh của thuật toán này lại phụ thuộc vào bản chất hàm một chiều của bài toán phân tích ra thừa số nguyên tố. Nếu có kỹ thuật mới giải bài toán này được phát hiện thì an ninh của thuật toán, cũng như PGP sẽ bị phá vỡ. Tương tự như vậy, thuật toán khóa đối xứng trong PGP là IDEA cũng có thể gặp phải những vấn đề về an ninh trong tương lai. Những phiên bản PGP gần đây hỗ trợ thêm những thuật toán khác nữa; vì thế mức độ an toàn trước sự tấn công về mặt mật mã học cũng thay đổi.

Vì rằng các tổ chức nghiên cứu lớn về mật mã học (như NSA, GCHQ...) không công bố những phát hiện mới của mình nên có thể tồn tại những phương pháp giải mã những thông điệp PGP mà không cần biết đến khóa bí mật được sử dụng. Điều này cũng đúng với bất kỳ hệ thống mật mã nào khác không chỉ là PGP.

Hiện nay PGP cho phép sử dụng một số thuật toán khác nhau để thực hiện việc mã hóa. Vì thế các thông điệp mã hóa với PGP hiện tại không nhất thiết có những điểm yếu giống như PGP phiên bản đầu. Tuy nhiên cũng có một số tin đồn về sự không an toàn của PGP phiên bản đầu tiên (sử dụng các thuật toán RSA và IDEA).

Phil Zimmermann, tác giả của PGP, đã từng bị chính phủ Hoa Kỳ điều tra trong vòng 3 năm về việc vi phạm những quy chế trong xuất khẩu phần mềm mật mã. Quá trình điều tra đã được kết thúc một cách đột ngột. Zimmermann cũng từng tuyên bố rằng sở dĩ chính phủ Hoa Kỳ kết thúc điều tra là vì họ đã tìm ra cách phá vỡ PGP trong thời kỳ đó.

Từ những lập luận ở trên, có thể khẳng định tương đối chắc chắn rằng tại thời điểm hiện tại chỉ những cơ quan thuộc về chính phủ mới có đủ những nguồn lực cần thiết để có thể phá vỡ những thông điệp PGP. Đối với tấn công phân tích mật mã từ phía cá nhân thì PGP vẫn tương đối an toàn.

5.4.4. Vài nét lịch sử

Phil Zimmermann tạo ra phiên bản PGP đầu tiên vào năm 1991. Vào thời điểm này, ông ta đã là một nhà hoạt động chống năng lượng hạt nhân và mục đích tạo PGP là để phục vụ những người có mục tiêu tương tự có thể sử dụng các hệ thống bảng thông báo điện tử (*bulletin board*) và lưu trữ tệp một cách an toàn. Đối với mục tiêu sử dụng phi thương mại, PGP hoàn toàn miễn phí và toàn bộ mã nguồn được bao gồm trong tất cả sản phẩm. PGP dễ dàng thâm nhập vào Usenet và từ đó vào Internet.

Tên gọi "*Pretty Good Privacy*" (tạm dịch: *Bí mật tương đối tốt*) được đặt theo tên của một cửa hiệu tạp hóa ở thành phố giả tưởng Lake Wobegon trong chương trình phát thanh của tác giả Garrison Keillor. Trong chương trình này, tên của hiệu tạp hóa là "*Ralph's Pretty Good Grocery*" (Tiệm tạp hóa tương đối tốt của Ralph).

Từ khi mới xuất hiện, PGP đã gặp rào cản về chính sách hạn chế xuất khẩu phần mềm mật mã của chính phủ Hoa Kỳ.

Ngay sau khi xuất hiện, PGP đã thu hút được khá nhiều người sử dụng trên Internet. Những người sử dụng và ủng hộ bao gồm những người bất đồng quan điểm tại những nước chuyên chế, những người bảo vệ quyền tự do cá nhân và những người ủng hộ tự do thông tin (*cypherpunk*).

Không lâu sau khi ra đời, PGP đã được sử dụng bên ngoài Hoa Kỳ và vào tháng 02 năm 1993, Zimmermann trở thành mục tiêu của một cuộc điều tra của chính phủ Hoa Kỳ về việc xuất khẩu "vũ khí" không có giấy phép. Tại thời điểm đó, các hệ thống mật mã với khóa lớn hơn 40 bit được xếp hạng cùng với vũ khí trong khi PGP chưa bao giờ sử dụng khóa có độ dài nhỏ hơn 128 bit. Mức hình phạt cho tội nói trên khá nặng nhưng cuộc điều tra đã đột ngột dừng lại mà không có một lời kết tội nào.

Chính sách hạn chế xuất khẩu mật mã vẫn còn hiệu lực nhưng đã được nới lỏng rất nhiều kể từ thập kỷ 1990. Từ năm 2000 trở đi thì việc tuân thủ các chính sách này không còn là điều khó khăn nữa. PGP không còn được xếp là vũ khí không được phép xuất khẩu và được phép xuất khẩu tới bất kỳ nơi nào nếu không bị cấm tại nơi đó.

Bằng sáng chế

Các phiên bản PGP đầu tiên còn gặp phải vấn đề về bằng sáng chế. Phiên bản đầu tiên sử dụng một thuật toán mã hóa khóa đối xứng do chính Zimmermann thiết kế (có tên là *Bass-O-Matic*). Ngay sau đó, ông ta thấy được thuật toán này không đảm bảo an ninh và thay thế bằng IDEA. Cả hai thuật toán RSA và IDEA đều đã được cấp bằng sáng chế và đòi hỏi có bản quyền để sử dụng. Đã có những tranh cãi khá gay gắt về việc Zimmermann sử dụng RSA và IDEA trong phần mềm của mình. Zimmermann tuyên bố rằng RSA Data Security (nay là RSA Security) đã cho phép (bằng lời nói) đối với việc sử dụng cho

các mục đích phi thương mại nhưng RSA không chính thức thừa nhận việc này. Cuộc điều tra đã bắt đầu từ đơn kiện của RSA DSI tới hải quan Hoa Kỳ về việc sử dụng thuật toán RSA trong PGP.

Vấn đề còn trở nên phức tạp hơn do tình trạng về bản quyền khác nhau ở các quốc gia. RSA chỉ được cấp bằng sáng chế tại Hoa Kỳ; những người nắm bản quyền IDEA tỏ ra rộng rãi hơn so với RSA. Thêm vào đó, bản quyền của thuật toán RSA được kiểm soát một phần bởi MIT thông qua RSA DSI (RSA Data Security Inc.). MIT không phản đối PGP nhưng các tác giả PGP vẫn gặp khó khăn do thái độ thù địch của RSA DSI đối với các sử dụng phi thương mại của RSA.

Tranh chấp về bản quyền RSA được giải quyết bằng việc phát triển PGP theo 2 nhánh:

- Phiên bản sử dụng trong Hoa Kỳ: sử dụng thư viện RSAREF (shareware của RSA).
- Phiên bản quốc tế: PGP-i, sử dụng mã RSA nguyên gốc của Zimmermann. Điều này cũng giúp tránh những khó khăn với quy chế hạn chế xuất khẩu phần mềm mật mã. PGP-i được duy trì và phân phối bởi Stale Schumacher ở Na Uy.

Phiên bản dành cho Hoa Kỳ được phân phối bởi nhiều nhà cung cấp, trong đó bao gồm cả MIT trên Internet, BBS, các cá nhân và nhóm người dùng. Ít nhất là trên website của MIT, để nhận được PGP thì địa chỉ E-mail phải thuộc về Hoa Kỳ hoặc Ca-na-đa. Bên ngoài Hoa Kỳ, người sử dụng tải về từ trang web của Schumacher: http://pgp.org.

Phát triển trong giai đoạn sau

Ngay trong thời gian tranh chấp, đội ngũ phát triển của Zimmermann tiếp tục đưa ra phiên bản PGP 3. Phiên bản này có nhiều cải thiện về an ninh, trong đó cấu trúc mới của chứng thực đã được sửa vài lỗi nhỏ của phiên bản 2.x cũng như cho phép dùng các

khóa khác nhau cho quá trình mã hóa và ký xác nhận. Bên cạnh đó, xuất phát từ bài học về bản quyền và xuất khẩu, PGP 3 đã loại bỏ hoàn toàn bản quyền. PGP 3 sử dụng thuật toán mật mã khóa đối xứng CAST-128 (còn gọi là CAST5) và thuật toán mật mã hóa khóa bất đối xứng DSA và ElGamal. Các thuật toán này đều không bị ràng buộc bởi bản quyền.

PGP thương mại

Như đã nêu ở phần trước, ngay từ khi xuất hiện, PGP đã gặp phải rắc rối trong vấn đề xuất khẩu ra ngoài Hoa Kỳ. Sau khi cuộc điều tra của chính phủ kết thúc vào năm 1996, Zimmermann và các cộng sự của mình khởi sự thành lập một công ty để phát triển các phiên bản mới của PGP. Công ty này được sát nhập với công ty Viacrypt (công ty đã mua bản quyền thương mại của PGP cũng như có bản quyền sử dụng RSA từ RSA DSI) và đổi tên thành tập đoàn PGP. Công ty mới thành lập này bắt đầu phát triển các phiên bản PGP mới dựa trên PGP 3. Khác với PGP 2 là phần mềm dựa trên dòng lệnh (*command line*), PGP được thiết kế từ đầu là một thư viện hàm cho phép người dùng có thể làm việc trên dòng lệnh cũng như thông qua môi trường đồ họa (*GUI*).

Thỏa thuận ban đầu giữa Viacrypt và Zimmermann là Viacrypt tạo ra các phiên bản đánh số chẵn còn Zimmermann các phiên bản đánh số lẻ. Trên cơ sở đó, Viacrypt tạo ra phiên bản PGP 4 dựa trên PGP 2. Để tránh nhầm lẫn về việc PGP 3 là phiên bản tiền thân của PGP 4 thì PGP 3 được đổi tên là PGP 5 và được tung ra vào tháng 5/1997.

Ngay tại PGP Inc., vẫn có mối lo ngại về vấn đề bản quyền. RSA DSI đưa ra phản đối về bản quyền sử dụng RSA đối với công ty mới thành lập. PGP Inc chuyển sang sử dụng một tiêu chuẩn nội bộ gọi là "*Unencumbered PGP*": không sử dụng bất kỳ một thuật toán nào bị ràng buộc bởi bản quyền.

OpenPGP và các phần mềm dựa trên PGP

Do tầm ảnh hưởng lớn của PGP trên phạm vi thế giới (được xem là hệ thống mật mã chất lượng cao được sử dụng nhiều nhất), rất nhiều nhà phát triển muốn các phần mềm của họ làm việc được với PGP 5. Đội ngũ phát triển PGP đã thuyết phục Zimmermann và đội ngũ lãnh đạo của PGP Inc. rằng một tiêu chuẩn mở cho PGP là điều cực kỳ quan trọng đối với công ty cũng như cộng đồng sử dụng mật mã. Ngay từ năm 1997 đã có một hệ thống tuân thủ theo các tiêu chuẩn của PGP của một công ty Bỉ tên là Veridis (lúc đó có tên là Highware) với bản quyền PGP 2 nhận được từ Zimmermann.

Vì vậy vào tháng 7 năm 1997, PGP Inc. đề xuất với IETF về một tiêu chuẩn mở có tên là OpenPGP. PGP Inc. cho phép IETF quyền sử dụng tên OpenPGP cho tiêu chuẩn cũng như các chương trình tuân theo tiêu chuẩn mới này. IETF chấp thuận đề xuất và thành lập nhóm làm việc về OpenPGP.

Hiện nay, OpenPGP là một tiêu chuẩn Internet và được quy định tại RFC 2440 (tháng 7 năm 1998). OpenPGP vẫn đang trong giai đoạn phát triển và quy định tiếp theo của RFC 2440 đang được nhóm làm việc tiếp tục hoàn thiện (vào thời điểm tháng 1 năm 2006).

Quỹ phát triển phần mềm tự do

Quỹ phát triển phần mềm tự do (*Free Software Foundation*) cũng phát triển một chương trình tuân theo OpenPGP có tên là *GNU Privacy Guard* (GnuPG). GnuPG được phân phối miễn phí cùng với mã nguồn theo giấy phép GPL. Ưu điểm của việc sử dụng GnuPG so với PGP (tuy GnuPG chưa có giao diện GUI cho Windows) là nó luôn được cung cấp miễn phí theo giấy phép GPL. Điều này đặc biệt quan trọng nếu người sử dụng muốn giải mã những tài liệu mã hóa tại thời điểm hiện nay trong một tương lai xa. Điều tương tự không đúng với PGP vì không có gì đảm bảo PGP sẽ được cung cấp miễn phí trong

tương lai. Trên thực tế, đối với PGP 9 thì phí bản quyền đã tăng ít nhất cho những người sử dụng PGP Personal; thêm vào đó, lịch sử phức tạp của bản quyền PGP cũng gây ra nhiều lo lắng.

Ngoài ra, nhiều nhà cung cấp khác cũng phát triển các phần mềm dựa trên OpenPGP.

Các phiên bản PGP xuất hiện sau khi có tiêu chuẩn vẫn tuân theo hoặc hỗ trợ OpenPGP.

Vào tháng 12 năm 1997, PGP Inc. được Network Associates Inc. (NAI) mua lại. Zimmermann và đội ngũ phát triển PGP trở thành nhân viên của NAI. NAI tiếp tục việc đi tiên phong trong việc xuất khẩu với chính sách xuất bản phần mềm (công ty đầu tiên có chính sách xuất khẩu bằng việc công bố mã nguồn). Dưới sự bảo hộ của NAI, đội ngũ PGP đã bổ sung các tính năng như mã hóa ổ đĩa, tường lửa, phát hiện xâm nhập và IPsec VPN vào họ các sản phẩm PGP.

Năm 2000, sau khi chính sách xuất khẩu phần mềm được thay đổi và không còn đòi hỏi việc công bố mã nguồn, NAI ngừng xuất bản mã nguồn của mình bất chấp sự phản đối của đội ngũ phát triển PGP. Việc này đã gây ra sự kinh ngạc cho người sử dụng PGP trên toàn thế giới.

Đầu năm 2001, Zimmermann bỏ việc tại NAI. Sau đó, ông ta giữ vai trò lãnh đạo về mật mã cho Hush Communications, một nhà cung cấp dịch vụ thư điện tử dựa trên OpenPGP. Ông ta cũng làm việc với Verisdis và một số công ty khác.

Tháng 10/2001, NAI tuyên bố bán các tài sản liên quan tới PGP và ngừng công việc phát triển PGP. Phần duy nhất được giữ lại là PGP E-Business Server (nguyên gốc là PGP Commandline). Tháng 2 năm 2002, NAI ngừng mọi hỗ trợ cho PGP trừ phần được giữ lại nói

trên. NAI (giờ đây là McAfee) tiếp tục bán và hỗ trợ sản phẩm này dưới tên là *McAfee E-Business Server*.

Tháng 8 năm 2002, một số thành viên cũ của đội ngũ phát triển PGP thành lập Tập đoàn PGP (PGP Corporation) và mua lại các tài sản liên quan tới PGP từ NAI. PGP Corp tiếp tục hỗ trợ những người sử dụng PGP và tôn trọng các hợp đồng hỗ trợ còn hiệu lực. Zimmermann trở thành cố vấn đặc biệt và nhà tư vấn cho PGP Corp đồng thời vẫn tiếp tục các mối quan hệ tại Hush Communications và Veridis cũng như điều hành công ty tư vấn riêng của mình.

NAI vẫn giữ bản quyền phiên bản dòng lệnh của PGP và tiếp tục bán ra dưới tên là "McAfee E-Business Server."

Cho tới trước tháng 01 năm 2004, theo thỏa thuận đã ký với NAI, PGP Corp không được quyền cung cấp phiên bản dòng lệnh của PGP. Tới giữa năm 2004, PGP Corp bắt đầu cung cấp sản phẩm này.

Với sự hợp tác của Zimmermann, Veridis phát triển và bán một phiên bản dòng lệnh tương thích với OpenPGP có tên là Filecrypt. Filecrypt và GnuPG được cung cấp đầy đủ mã nguồn cũng như cung cấp các phiên bản trước đó trên nhiều nền tảng khác nhau.

Sau khi mua lại tài sản liên quan tới PGP từ NAI (2002), PGP Corp cung cấp hỗ trợ kỹ thuật về PGP trên toàn thế giới.

5.4.5. Các phiên bản của PGP Corp. Theo thứ tự thời gian:

2002

- PGP 7.2 cho Mac OS 9.
- PGP Personal và PGP Freeware.
- PGP 8.0 cho Macintosh và Windows.
- PGP Corporation công bố mã nguồn.

2003

- PGP Desktop 8.0.1DE cho Windows tiếng Đức.
- PGP Desktop 8.0.2.
- PGP Desktop 8.0.3 cho Macintosh và Windows.
- Công bố và đóng gói PGP Universal INFO, một dòng sản phẩm mới.
- PGP Universal 1.1 (30 tháng 12).

2004

- PGP Universal 1.2.
- PGP Desktop 8.1.
- PGP Command Line 8.5.
- PGP Corporation và Symantec đưa ra giải pháp an ninh thư điện tử tích hợp PGP Universal cho doanh nghiệp.
- PGP Software Development Kit (SDK) nhận được FIPS 140-2 Level 1 từ NIST.

2005

- PGP Universal 2.0 và PGP Desktop 9.0 cũng như dịch vụ PGP Global Directory.
- "Tiger" cho Mac OS X 10.4 .
- Nâng cấp PGP 9.0.1 Freeware thành bản đầy đủ tính năng dưới dạng phần mềm dùng thử 30 ngày.
- PGP Whole Disk Encryption được chính thức phát hành như một sản phẩm độc lập.
- PGP 9.0.2 với phần cập nhật cho bản chuyển mã quốc tế và bản địa hóa tiếng Đức.
- PGP 9.0.2 với phần cập nhật cho bản địa hóa tiếng Nhật.

Sự tương thích giữa các phiên bản PGP

Các vấn đề về bản quyền và chính sách xuất khẩu đã gây ra một số vấn đề tương thích giữa các phiên bản PGP. Tuy nhiên từ khi OpenPGP được chấp thuận và từ khi Tập đoàn PGP được thành lập (2002) thì tình trạng nói trên đã được cải thiện đáng kể.

OpenPGP quy định các cơ chế thương lượng giữa các chương trình PGP ở các phía của đường truyền cũng như thuật toán mã hóa được sử dụng và các tính năng bổ sung khác từ phiên bản PGP 2.x. Tất cả các chương trình tuân theo PGP đều bắt buộc phải thực hiện những quy định này. Vì vậy, không tồn tại những vấn đề tương thích lớn giữa các phiên bản PGP, bất kể nó được lập trình từ đâu: PGP Corp, McAfee, Gnu/FSF (ie, GPG), Hushmail, Veridis, Articsoft, Forum... Các lập trình viên của các chương trình này cũng có mối quan hệ nhất định với nhau. Họ coi những bất tương thích là các lỗi phần mềm và sửa mỗi khi phát hiện ra.

Tương thích của PGP 2.x

Khả năng tương thích của các phiên bản PGP gần đây với PGP 2.x có phần phức tạp hơn. PGP 2 đã sử dụng các thuật toán có bản quyền dưới nhiều điều khoản khác nhau. Bản quyền của RSA đã hết hiệu lực từ năm 2000 nhưng bản quyền của IDEA chỉ hết hiệu lực vào 2010-2011.

Một số phiên bản PGP gần đây cung cấp khả năng tương thích với PGP 2.x (các phiên bản của PGP Corp và Hushmail) nhưng các phiên bản của các nhà cung cấp khác thì không tương thích. Đáng kể nhất là GnuPG không đảm bảo tính năng này (IDEA). Để có thể làm việc với PGP 2.x thì phải có mô-đun bổ sung (plug-in) cho GnuPG. Tuy nhiên người dùng phải tự xây dựng mô-đun này. Để sử dụng IDEA cho các mục đích thương mại thì người dùng cần phải có giấy phép trong khi họ có thể sử dụng miễn phí cho các mục đích khác.

Vào thời điểm năm 2004, cách tốt nhất để tránh các vấn đề không tương thích với PGP 2.x là không sử dụng chúng và sử dụng các phiên bản tuân theo chuẩn OpenPGP.

Một số vấn đề nhỏ về an ninh của PGP 2.x đã được phát hiện và một số đã được sửa. Tuy nhiên một số trong các giao thức cơ bản dùng trong PGP 2.x có những điểm yếu có thể bị tấn công và chúng vẫn chưa được sửa. Các lỗi này không xuất hiện trong tiêu chuẩn OpenPGP cũng như các bản thực hiện tiêu chuẩn.

Mặc dù các bản PGP 2.x đã vá lỗi không có vấn đề nghiêm trọng nào nhưng nhóm làm việc của IETF vẫn không tán thành việc tương thích với OpenPGP. Staale Schumacher Ytteborg vẫn duy trì trang web *pgpi.org* trong đó cung cấp hầu hết các phiên bản PGP kể từ 2.x.

Do nguyên nhân lịch sử, giữa các phiên bản PGP 2.x tồn tại vấn đề không tương thích một cách chủ ý (do bản quyền RSA). Một phần trong những nỗ lực giải quyết điều này là yêu cầu của phiên bản 2.6 phải tương thích với các phiên bản 2.x trước nó. Điều này được thực hiện bằng cách nâng cấp cấu trúc dữ liệu bên trong và sử dụng bản thực hiện RSAREF của RSA.

Mã nguồn của PGP thực hiện thuật toán RSA có thể được sử dụng hợp pháp bên ngoài Hoa Kỳ (chẳng hạn PGP 2.6.3i). Bộ mã này có tốc độ thực hiện thuật toán nhanh gấp đôi so với mã của RSAREF.

Trong thời điểm đó, tại Hoa Kỳ, đội ngũ phát triển PGP đã viết PGP 3 (sau này đổi tên thành PGP 5, xem phần trên) và tiêu chuẩn OpenPGP đã được chấp nhận. Các khó khăn về bản quyền đã buộc họ phải loại bỏ RSA nhưng vào năm 2000 (khi bản quyền hết hạn) thì PGP và OpenPGP tiếp tục hỗ trợ thuật toán này. Và từ đó không tồn tại các phiên bản cho Hoa Kỳ và quốc tế riêng biệt nữa.

Tóm lại, trong thời điểm hiện nay, người sử dụng nên dùng các phiên bản mới tuân theo OpenPGP. Sự hợp tác giữa các nhà phát triển đã giải quyết phần lớn các vấn đề không tương thích giữa chúng.

- So sánh đặc tính của các phiên bản. So sánh với RFC 1991 (PGP 2.x), OpenPGP đưa ra nhiều tính năng mới. Nó hỗ trợ khả năng tương thích ngược, có nghĩa là các phiên bản thực hiện OpenPGP có thể đọc và sử dụng các khóa, chứng thực của các phiên bản trước đó.
- PGP 2.x không có khả năng tương thích xuôi vì nó không thể sử dụng các văn bản hay khóa tuân theo OpenPGP.

Trong bảng sau, các thuật toán bắt buộc được đánh dấu bằng dấu *.

Đặc tính	PGP 2.x (RFC 1991)	OpenPGP (RFC 2440)
Định dạng khóa	Khóa v3	Khóa v4
Thuật toán khóa bất đối xứng	*RSA (mã hóa & chữ ký)	RSA (mã hóa và chữ ký) *DSA (chữ ký) *Elgamal (mã hóa)
Thuật toán khóa đối xứng	*IDEA	IDEA *Triple-DES CAST5 Blowfish AES 128, 192, 256 Twofish
Hàm băm mật mã	*MD5	MD5 *SHA-1 RIPEMD-160 SHA-256 SHA-384 SHA-512
Thuật toán nén	ZIP	ZIP gzip bzip2

Các tính năng bổ sung của khóa v4 so với v3 của OpenPGP:

- Khóa công khai có thể có các khóa con bên cạnh khóa chính, cho phép sử dụng các khóa khác nhau cho mã hóa và chữ ký.

- Hỗ trợ nhiều thuật toán khác nhau để đảm bảo khả năng tương thích:

+ Một số thuật toán là bắt buộc

+ Khóa công khai của người nhận có thể xác định thứ tự ưu tiên của các thuật toán

- Mô hình *mạng lưới tín nhiệm* được mở rộng với khả năng hỗ trợ tính năng *chữ ký được tin tưởng* (chữ ký này không những được tin mà còn được quyền xác nhận những chữ ký khác), cho phép thực hiện một dạng của *cơ quan cấp chứng thực số*.

- Một chứng thực số có thể quy định một khóa khác có khả năng thu hồi nó.

- Một số lỗi an ninh nhỏ trong mô tả ID và định dạng được sửa.

(v3 và v4 chỉ đến hệ thống phiên bản sử dụng bên trong định dạng dữ liệu chứ không phải phiên bản phần mềm PGP)

5.4.6. Các phần mềm thực hiện

Sau đây là danh mục một số phần mềm thực hiện các phiên bản của PGP.

- Authora Inc. Thành viên sáng lập của Open PGP Alliance - Người tạo ra Zendit (phần mềm PGP mở để bàn tự do dành cho cá nhân) và EDGE (Dòng lệnh mở PGP).

- McAfee Inc. - McAfee E-Business Server – Dòng lện PGP gốc dành cho Windows, Solaris, AIX, LINUX, HPUX, và máy tính lớn (OS/390, z/OS).

- PGP Corporation - nhà giám sát, nhà cung cấp và nhà hỗ trợ hiện tại của PGP văn phòng.

- GNU Privacy Guard (aka GnuPG hoặc GPG).
- WinPT, giao diện đồ họa.
- GPGee, mở rộng Windows explorer dành cho GnuPG.
- GPGshell, nguồn vào Windows dành cho GnuPG.
- Enigmail, sự mở rộng E-mail đối với họ Mozilla.
  + MacGPG, the Mac OS X port of GnuPG.
  + GPGMail, a plugin for Apple's Mail
  + KGPG - a simple, KDE frontend for GnuPG.
  + GPGol - a plugin for Microsoft Outlook 2003.
  + Gpg4win - a windows bundle of WinPT, GPGee, GPA, GPGol, and more
- Patrick Townsend & Associates là một công ty thương mại đầu tiên để đưa GPP đến hệ điều hành IBM os/400 iSeries.
- EasyByte Cryptocx - OpenPGP tương thích với thành phần DLL.
- Veridis - phiên bản dòng lệnh của PGP.
- BSD Privacy Guard - BSD cấp giấy phép thực hiện PGP, bắt đầu bởi NetBSD cùng sự trợ giúp của Google Summer of Code.
- PGPfreeware 6.0.2i
- Danh sách Website PGP bằng tiếng Anh
- PGPfreeware 7.0.3 for Windows (sử dụng cho mục đích phi thương mại).
- Hướng dẫn cho người mới bắt đầu học PGP 6.5.8
- PGP 6.5.8 for Inix (phần mềm miễn phí)

6

MỘT SỐ GIAO THỨC BẢO MẬT THÔNG DỤNG KHÁC

Ngoài những vấn đề bảo mật trong quốc phòng, an ninh… một trong những loại giao dịch điện tử phổ biến rộng rãi trong xã hội có yêu cầu bảo mật rất cao là những giao dịch thương mại, nhất là vấn đề thanh toán trong Thương mại điện tử. Các giao dịch đó thực chất đều là việc trao đổi những thông điệp có chứa thông tin cần được bảo mật (thư trao đổi, hợp đồng, thanh toán tiền, v.v.).

Sau đây ta lần lượt xét đến một số giao thức bảo mật sử dụng phổ biến hiện nay trong giao dịch điện tử, chủ yếu là trong các dịch vụ Internet và thông tin thanh toán trong thương mại điện tử.

Các hệ thống mật mã hiện nay đang được sử dụng phổ biến nói chung có thể chia làm hai nhóm chính.

Nhóm thứ nhất bao gồm các chương trình và giao diện được sử dụng trong mã hóa dữ liệu trong các thư điện tử: các chương trình đọc các thông điệp trong thư điện tử và lưu giữ dưới dạng mật mã hoặc chuyển cho đối tác đã được cấp khóa mã như là S/MIME.

Các chương trình này cũng được sử dụng cho một người (*single user*) để tự bảo vệ các tệp lưu giữ trên máy tính cá nhân của mình.

Nhóm thứ hai là các hệ thống giao diện mạng được sử dụng với mục đích cung cấp các tính năng như bảo mật, xác nhận, đồng bộ

hóa và lọc thông tin trong môi trường mạng. Các hệ thống này đều yêu cầu phản hồi tức thời giữa từng người dùng trong hệ thống khách hàng với một máy chủ có cấu hình chuẩn để hoạt động đúng quy cách. Nhiều hệ thống trong nhóm này đã trở thành công cụ nền tảng cho các website thương mại điện tử như: SSL, PCT. S-HTTP, SET, và SSH…

6.1. GIAO THỨC BẢO MẬT THƯ ĐIỆN TỬ MỞ RỘNG ĐA PHƯƠNG TIỆN

PGP cũng là một giao thức được sử dụng bảo mật có hiệu quả cho dịch vụ thư điện tử. Tuy vậy do bởi các công ty cung cấp dịch vụ hòm thư điện tử đều có quan hệ kinh doanh rất chặt chẽ với RSA Data Security nên S/MIME thường dùng phổ biến cho các hòm thư điện tử hơn là PGP.

6.1.1. Giao thức mở rộng thư điện tử đa phương tiện trên Internet có bảo mật (S/MIME)

Giao thức mở rộng thư điện tử đa phương tiện trên Internet - có bảo mật S/MIME (Secure/Multipurpose Internet Mail Extension) là một chương trình do RSA Data Security thiết kế giống như một hộp công cụ mã hóa cho phép gắn chữ ký số của người gửi vào các tin đính kèm trong hộp thư mở rộng đa phương tiện sử dụng giao thức MIME (Multipurpose Internet Mail Extension). MIME được mô tả trong giao thức RFC 1521 và được đề xuất sử dụng làm chuẩn chính thức cho thư điện tử mở rộng, tức là sử dụng cho việc truyền tải các tệp đính kèm multimedia trong hộp thư điện tử…

Để gửi tệp đính kèm thư điện tử cần được bảo vệ cho một đối tác, cả hai hòm thư đều phải đăng ký sử dụng S/MIME và người gửi phải được cung cấp khóa công khai của người nhận.

S/MIME[(1)] đã được tiêu chuẩn hóa chuyển thành IETF và đã xuất hiện nhiều công bố mô tả S/MIME phiên bản thứ ba. Hiện thời

[(1)] Thông tin chi tiết về MIME có thể tìm được tại địa chỉ: ftp://ftp.isi.edu/in-notes /rfc1521.txt

S/MIME được gắn với một số nhà cung cấp dịch vụ mạng và dịch vụ thư điện tử hàng đầu như: ConnectSoft, Frontier, FTP Software, Qualcomm, Microsoft, Lotus, Wollongong, Banyan, NCD, SecureWare, VeriSign, Netscape, và Novell.

6.1.2. Chức năng

S/MIME cung cấp những dịch vụ mã hóa bảo mật sau đây cho các ứng dụng truyền thông điệp: Nhận dạng, toàn vẹn thông tin và chống chối bỏ của người phát hành thông điệp (sử dụng chữ ký số) cũng như bí mật và an toàn dữ liệu (dùng mật mã hóa). S/MIME được dùng đặc biệt cho các ứng dụng thông điệp mở rộng đa phương tiện (MIME) kiểu *application/pkcs7-mime* (kiểu smime "dữ liệu được bọc") nhằm mã hóa dữ liệu trong đó toàn bộ thực thể MIME được bao bọc và đóng gói thành một đối tượng, đối tượng này tiếp đó được chèn vào một thực thể *application/pkcs7-mime MIME.*

Một thông điệp thư điện tử gồm hai phần: phần tiêu đề (*header*) và phần nội dung hay phần thân (*body*). Cấu trúc của phần tiêu đề có thể tìm thấy trong giao thức RFC 822. Cấu trúc của phần thân thường không xác định sẵn ngoại trừ trường hợp thư điện tử được sử dụng định dạng MIME. MIME quy định cấu trúc mặc định của phần thân thư điện tử, cho phép thư điện tử bao gồm những phần văn bản tăng cường, hình ảnh, âm thanh… được tiêu chuẩn hóa thông qua các hệ thống thư MIME. MIME cho phép các hệ thống E-mail tích hợp được thông tin dữ liệu dạng văn bản, hình ảnh và âm thanh tuy nhiên bản thân MIME không cung cấp dịch vụ bảo mật. Nội dung của giao thức S/MIME chính là xác định những dịch vụ bảo mật cần thiết, tuân theo cú pháp trong PKCS#7 cho chữ ký số và thuật toán mã hóa. Phần thân của MIME mang một thông điệp PKCS#7, bản thân nó là kết quả mã hóa trên một phần thân của MIME.

6.1.3. Các chứng thư S/MIME

Trước khi S/MIME được dùng cho một trong các ứng dụng nói ở mục trên, chủ hòm thư cần phải nhận được và phải cài đặt một khóa

kèm theo chứng thư cá nhân do một cơ quan chứng thực số nội bộ hoặc do một cơ quan chứng thực số công cộng cấp. Thực tế nhất là nên dùng những khóa bí mật (và những chứng thư kèm theo) riêng rẽ cho việc sử dụng chữ ký và cho việc mã hóa vì điều này cho phép bạn trao đổi khóa mã hóa mà không làm ảnh hưởng lộ bí mật về chữ ký. Thuật toán mã hóa đòi hỏi trong kho dữ liệu lưu trữ của bạn phải có chứng thư của đối tác nhận thông điệp của bạn (việc lưu trữ này là hoàn toàn tự động hóa mỗi khi bạn nhận được thông điệp từ một đối tác có kèm một chữ ký có giá trị hợp lệ). Về mặt công nghệ, bạn hoàn toàn có thể gửi một thông điệp mã hóa (sử dụng chứng thư của người nhận thư) dù rằng bạn không có chứng thư về chữ ký của mình, tuy nhiên trong thực tế các khách sử dụng S/MIME bao giờ cũng yêu cầu bạn cài đặt chứng thư của chính bạn trước khi họ cho phép bạn sử dụng khóa mã của họ.

Một chứng thư cá nhân *cơ bản* ("lớp thứ nhất") chỉ có thể kiểm tra để xác thực "căn cước" người gửi, xem thử người đã gửi E-mail có thực sự là chủ nhân của địa chỉ ghi ở ô "From:" trong E-mail đã nhận được hay không theo nghĩa là người đã gửi E-mail đến cho bạn có thể nhận được những thư trả lời gửi đến địa chỉ ghi trong ô "From" đó hay không. Chứng thư lớp cơ bản này không cho phép bạn kiểm tra được tên và doanh nghiệp của người đã gửi E-mail. Muốn biết được điều này, bạn cần đòi hỏi một chứng thư "lớp thứ hai" từ một CA có lưu trữ và xác nhận những thông tin chi tiết hơn của người được cấp chứng thư.

Tùy thuộc vào chính sách của từng CA, có những CA quy định niêm yết công khai thông tin của người được cấp chứng thư để phục vụ cho việc tìm kiếm và kiểm tra trong khi nhiều CA khác lại không cung cấp các thông tin cá nhân cụ thể như tên, doanh nghiệp công tác mà chỉ cung cấp những thông tin tối thiểu như số chứng thư (*serial)* và danh sách các chứng thư bị thu hồi để bạn tự kiểm tra mà thôi.

6.1.4. Trở ngại khi triển khai S/MIME trong thực tế

Khi triển khai sử dụng S/MIME cho các ứng dụng trên Internet ta có thể gặp một số trở ngại sau đây.

- Không phải là phần mềm E-mail nào cũng tải vào đó được chữ ký của S/MIME, kết quả là tập tin đính kèm được gọi là *smime.p7s* có thể làm cho một số người bị nhầm lẫn.

- Nhiều khi S/MIME bị xem là không thực sự phù hợp cho khách sử dụng thông qua webmail. Dù rằng có thể có những trợ giúp chèn được vào trình duyệt nhưng có những dịch vụ thực hành bảo vệ vẫn đòi hỏi một khóa riêng để cho phía người dùng có thể truy cập còn từ phía máy chủ webmail thì không truy cập được: điều này gây phiền phức cho lợi thế của khóa webmail trong việc cung cấp khả năng truy cập một cách phổ biến. Điều này không riêng gì cho S/MIME, thực ra những biện pháp an toàn khác để ký (*signing*) một webmail cũng có thể đòi hỏi trình duyệt web (*browser*) phải mã hóa để tạo chữ ký, ngoại trừ PGP Desktop và một số phiên bản của GnuPG, các phần mềm này có thể tách dữ liệu ra khỏi webmail, ký bằng clipboard rồi chuyển lại dữ liệu đã ký vào trang webmail. Về mặt an toàn thì thực ra đấy lại là biện pháp tốt hơn.

- S/MIME được thiết kế riêng cho việc bảo vệ an toàn đầu-đến-cuối. Thuật toán mã hóa sẽ không chỉ mã hóa các thông tin của bạn gửi đi mà cũng mã hóa luôn cả các phần mềm độc (virus) nếu có. Vì vậy, nếu mail của bạn được quét các phần mềm độc ở khắp mọi nơi nhưng chỉ trừ các điểm cuối, chẳng hạn như các cổng kết nối của toàn công ty của bạn thì việc mã hóa sẽ vô hiệu hóa các máy quét và bản mail sẽ phát tán thành công các phần mềm độc.

Giải pháp khắc phục có thể là:

- Thực hiện quét mã độc tại đầu cuối máy trạm *sau khi* đã giải mã.

- Lưu trữ các khóa riêng trên máy chủ của cổng, như vậy thì việc giải mã có thể thực hiện trước khi quét mã độc. (Tuy nhiên về mặt

bảo mật thì biện pháp này không được tối ưu vì có thể cho phép vài kẻ nào đó truy cập vào máy chủ cổng để đọc thư của người khác!)

- Sử dụng bộ quét nội dung thông điệp được thiết kế đặc biệt cho việc quét nội dung của thông điệp đã mã hóa còn vẫn giữ nguyên các chữ ký và bản mã hóa. Giải pháp này phải chứa một công cụ bảo vệ được tích hợp, sử dụng cho cả khóa riêng dùng để giải mã thông điệp và cho cả phần nội dung tạm thời được giải mã.

6.2. AN NINH TẦNG GIAO VẬN VÀ TẦNG ĐỆM BẢO MẬT

6.2.1. SSL và TLS

SSL (Secure Socket Layer) là giao thức đa mục đích được thiết kế để tạo ra các giao tiếp giữa hai chương trình ứng dụng trên một cổng định trước (*socket 443*) nhằm mã hóa toàn bộ thông tin đi/đến, mà ngày nay được sử dụng rộng rãi cho giao dịch điện tử như truyền số hiệu thẻ tín dụng, mật khẩu, số bí mật cá nhân PIN (Personal Information Number) trên Internet, trên các thẻ tín dụng v.v.

Giao thức SSL được hình thành và phát triển bởi Netscape, và ngày nay đã được sử dụng rộng rãi trên World Wide Web trong việc xác thực và mã hóa thông tin giữa phía khách (*client*) và phía máy chủ (*server*). Tổ chức IETF (*Internet Engineering Task Force: Lực lượng công tác kỹ thuật về Internet*) đã chuẩn hóa SSL và đặt lại tên là TLS (*Transpot Layer Security: An ninh lớp giao vận*). Tuy nhiên SSL vẫn là thuật ngữ được sử dụng rộng rãi hơn.

SSL được thiết kế như là một giao thức riêng cho vấn đề bảo mật có thể hỗ trợ rất nhiều ứng dụng. Giao thức SSL hoạt động bên trên TCP/IP và bên dưới các giao thức ứng dụng tầng cao hơn như là HTTP (Hyper Text Transpot Protocol: Giao thức truyền tải siêu văn bản), IMAP (Internet Messaging Access Protocol: Giao thức truy nhập bản tin Internet) và FTP (File Transport Protocol: Giao thức truyền file). SSL có thể sử dụng để hỗ trợ các giao dịch an toàn cho

rất nhiều ứng dụng khác nhau trên Internet, và hiện nay SSL được sử dụng chính cho các giao dịch trên Web.

SSL không phải là một giao thức đơn lẻ, mà là một tập hợp các thủ tục đã được chuẩn hóa để thực hiện các nhiệm vụ bảo mật sau:

Xác thực Server

Cho phép người sử dụng xác thực được server muốn kết nối. Lúc này, phía browser sử dụng các kỹ thuật mã hóa công khai để chắc chắn rằng certificate và public ID của server là có giá trị và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy của client. Điều này rất quan trọng đối với người dùng. Ví dụ như khi gửi mã số credit card qua mạng thì có người dùng thực sự muốn kiểm tra liệu server sẽ nhận thông tin có đúng là server mà họ định gửi đến không.

Xác thực Client

Cho phép phía server xác thực được người sử dụng muốn kết nối. Phía server cũng sử dụng các kỹ thuật mã hóa công khai để kiểm tra xem certificate và public ID của server có giá trị hay không và được cấp phát bởi một CA trong danh sách các CA đáng tin cậy của server không. Điều này rất quan trọng đối với các nhà cung cấp. Ví dụ khi một ngân hàng định gửi các thông tin tài chính mang tính bảo mật tới khách hàng thì họ rất muốn kiểm tra định danh của người nhận. Mã hóa kết nối: Tất cả thông tin trao đổi giữa client và server được mã hóa trên đường truyền nhằm nâng cao khả năng bảo mật. Điều này rất quan trọng đối với cả hai bên khi có các giao dịch mang tính riêng tư. Ngoài ra, tất cả các dữ liệu được gửi đi trên một kết nối SSL đã được mã hóa còn được bảo vệ nhờ cơ chế tự động phát hiện các xáo trộn, thay đổi trong dữ liệu.

6.2.2. Hoạt động của SSL

Điểm cơ bản của SSL là nó được thiết kế độc lập với tầng ứng dụng để đảm bảo tính bí mật, an toàn và chống giả mạo luồng thông

tin qua Internet giữa hai ứng dụng bất kỳ, ví dụ như webserver và các trình duyệt khách (browsers), do đó được sử dụng rộng rãi trong nhiều ứng dụng khác nhau trên môi trường Internet.

Toàn bộ cơ chế hoạt động và hệ thống thuật toán mã hóa sử dụng trong SSL được phổ biến công khai, trừ khóa phiên chia sẻ tạm thời (*session key*) được sinh ra tại thời điểm trao đổi giữa hai ứng dụng là tạo ngẫu nhiên và bí mật đối với người quan sát trên mạng máy tính.

Ngoài ra, giao thức SSL còn đòi hỏi ứng dụng chủ phải được chứng thực bởi một đối tượng lớp thứ ba (CA) thông qua chứng thực điện tử (*digital certificate*) dựa trên mật mã công khai (ví dụ RSA).

Giao thức SSL dựa trên hai nhóm con giao thức là giao thức "bắt tay" (*handshake protocol*) và giao thức "bản ghi" (*record protocol*).

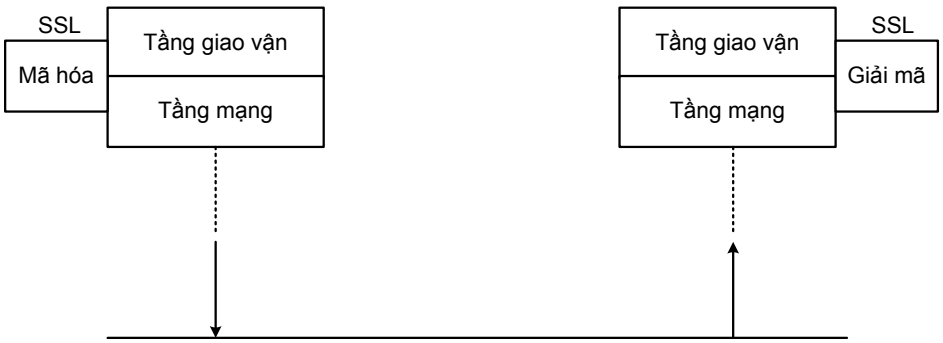
Giao thức bắt tay xác định các tham số giao dịch giữa hai đối tác có nhu cầu trao đổi thông tin hoặc dữ liệu, còn giao thức bản ghi xác định khuôn dạng cho việc tiến hành mã hóa và truyền tin hai chiều giữa hai đối tác đó. Khi hai ứng dụng máy tính, ví dụ giữa một trình duyệt web và máy chủ web, làm việc với nhau, máy chủ và máy khách sẽ trao đổi "lời chào" dưới dạng các thông điệp gửi cho nhau với xuất phát đầu tiên chủ động từ máy chủ, đồng thời xác định các chuẩn về thuật toán mã hóa và nén số liệu có thể được áp dụng giữa hai ứng dụng.

Ngoài ra, các ứng dụng còn trao đổi "số nhận dạng/khóa theo phiên" (*session ID, session key*) duy nhất cho lần làm việc đó. Sau đó ứng dụng khách (trình duyệt) yêu cầu có chứng thực điện tử (*digital certificate*) xác thực của ứng dụng chủ (web server). Chứng thực điện tử (chứng thư) thường được xác nhận bởi một cơ quan trung gian là CA như RSA Data Sercurity.., một dạng tổ chức độc lập, trung lập và có uy tín. Các tổ chức này cung cấp dịch vụ "xác nhận" số nhận dạng của một công ty và phát hành chứng chỉ duy nhất cho

công ty đó như là bằng chứng nhận dạng (*identity*) cho các giao dịch trên mạng, ở đây là các máy chủ (*web server*).

Sau khi kiểm tra chứng thư (chứng chỉ điện tử) của máy chủ (sử dụng thuật toán mật mã công khai, như RSA tại trình máy trạm), ứng dụng máy trạm sử dụng các thông tin trong chứng thư để mã hóa thông điệp gửi lại máy chủ mà chỉ có máy đó mới có thể giải mã. Trên cơ sở đó, hai ứng dụng trao đổi khóa chính (*master key*) (khóa bí mật hay khóa đối xứng) để làm cơ sở cho việc mã hóa luồng thông tin/dữ liệu qua lại giữa hai ứng dụng chủ khách.

TLS hoặc SSL có thể xem như một tầng giao thức trung gian giữa tầng mạng và tầng giao vận trong mô hình DoD (5 tầng) hoặc OSI (7 tầng) của mạng máy tính. Trong TLS hoặc SSL, mỗi thông điệp được chuyển đi cho một đối tác được cấp chứng nhận giao dịch hoặc nhận từ đối tác đó đều được mã hóa bởi một khóa đối xứng khi chuyển đi và được giải mã khi nhận đến, thông điệp đó còn được gắn một mật mã nhận dạng (có vai trò như một chữ ký điện tử) được hệ thống cấp cho mỗi đối tác. SSL sử dụng sự xác nhận thông qua mật mã chung X509.



Hình 6.1: SSL trong giao thức mạng

Một website sử dụng giao thức http được tích hợp SSL có tính năng bảo mật thông tin gửi từ phía máy khách (*client side*) vào trang web đến phía máy chủ (*server side*) vì thông tin ở tầng giao vận bên máy khách phải qua tầng phụ SSL để được mã hóa (theo luật mã hóa

công khai đã được SSL cung cấp cho máy chủ trang web) rồi mới quay về tầng mạng để tiếp tục chuyển đi: dữ liệu truyền đi trên môi trường Internet đã được mã hóa (*ciphertext*). Phía máy chủ khi dữ liệu về đến tầng mạng thì lại được đưa sang tầng phụ SSL để được giải mã (bằng khóa riêng của phía máy chủ tương ứng với khóa công khai trên trang web) rồi quay về tầng giao vận để chuyển xuống tầng áp dụng: thông tin tầng ứng dụng nhận được lại là thông tin tường minh (*plaintext*).

Giao thức http có tích hợp SSL thường ký hiệu là: https. Các trang web dùng cho dịch vụ ngân hàng trực tuyến của các website thanh toán điện tử an toàn nhất thiết đều cần được sử dụng giao thức này.

Công nghệ truyền thông riêng tư (PCT)

Công nghệ truyền thông riêng tư PCT (*Private Communication Technology*) PCT 1.0 cũng là một giao diện an toàn ở tầng giao vận (*transport layer)* được hãng Microsoft phát triển vào khoảng giữa những năm 1990 để khắc phục những lỗ thủng trong phiên bản 2.0 và để thúc ép hãng Nescape từ bỏ quyền kiểm soát SSL 2.0 mà lúc đó họ đang sở hữu bản quyền.

Về sau PCT được thay thế bởi SSLv3 và TLS. Trong một giai đoạn ngắn PCT còn được Internet Explorer hỗ trợ nhưng các phiên bản sau này không còn nữa. PCT hiện còn được thấy trong IIS và trong các thư viện hệ điều hành Windows tuy nhiên trong Windows Server 2003 thì mặc định là không cho phép sử dụng.

6.3. CÁC GIAO THỨC TRUYỀN THÔNG CÓ BẢO MẬT

6.3.1. HTTPS

Giao thức truyền thông siêu văn bản có bảo mật HTTPS (Hypertext Transfer Protocol Secure) là một tổ hợp của HTTP (Hypertext Transfer Protocol: giao thức truyền thông siêu văn bản)

với SSL/TLS để cung cấp dịch vụ truyền thông được mã hóa và nhận dạng an toàn cho một máy chủ web. Giao thức HTTPS thường được dùng cho các website thanh toán điện tử trên WWW hoặc cho các giao dịch nhạy cảm trong một hệ thống thông tin lớn.

Netscape Communications tạo ra HTTPS trong năm 1994 dùng cho trình duyệt web Netscape Navigator. Thoạt đầu, HTTPS được dùng với chuẩn mã hóa SSL nhưng sau đó SSL phát triển thành TLS cho nên phiên bản hiện nay của HTTPS được ký hiệu định danh là RFC 2818 vào hồi tháng 5 năm 2000.

Ý tưởng chính của HTTPS là tìm cách tạo ra một kênh truyền tin an toàn trên một mạng không an toàn. Điều này có thể cung cấp những phương thức bảo vệ có hiệu quả chống lại những kẻ "nghe lén" và chống lại sự tấn công của "kẻ đứng giữa" bằng cách dùng một dãy quy tắc mã hóa thích hợp và thiết kế sao cho chứng thư của máy chủ phải được kiểm tra và tin tưởng. "Niềm tin" tạo được trong HTTPS dựa chủ yếu vào cơ sở các cơ quan chứng thực điện tử (CA) được cài đặt trước trên trình duyệt. Do vậy, một sự kết nối HTTPS đến một website có thể được tin cậy khi và chỉ khi các điều kiện sau đây được thực hiện:

1. Người sử dụng tin tưởng rằng trình duyệt của họ thực hiện một cách đúng đắn giao thức HTTPS đã được cài đặt trước với những CA đáng tin cậy.
2. Người sử dụng tin tưởng là CA chỉ chứng thực cho những website hợp pháp, không có quan hệ với những website lừa đảo.
3. Website xuất trình một chứng thư hợp lệ, nghĩa là được ký xác nhận bởi một CA đáng tin cậy.
4. Trong chứng thư chỉ rõ căn cước nhận dạng của website (nghĩa là nếu trình duyệt truy cập đến địa chỉ: "https://vidu.com" thì chứng thư của website thực sự thuộc về công ty vidu chứ không phải thuộc về tổ chức khác!)

5. Hoặc là mọi can thiệp ngẫu nhiên trên Internet đều đáng tin cậy hoặc là người sử dụng tin tưởng là tầng mạng được mã hóa bởi giao thức bảo mật (TLS hay SSL) là không thể bị nghe lén.

Địa chỉ URL của các website thông thường dùng giao thức HTTP bắt đầu với cụm ký tự "http://" và mặc định sử dụng cổng 80. Các website sử dụng giao thức có bảo mật HTTPS có địa chỉ URL bắt đầu bởi cụm ký tự "https://" và sử dụng mặc định cổng 443.

Tầng mạng

HTTP hoạt động ngay ở tầng ứng dụng, tầng cao nhất trong mô hình tham chiếu OSI nhưng giao thức bảo mật thì lại hoạt động ở một tầng phụ thấp hơn: giao thức này mã hóa thông điệp trước khi gửi đi và giải mã thông điệp sau khi nhận được. Nói đúng ra, HTTPS không hẳn là một giao thức mà là dùng để chỉ việc sử dụng HTTP thông thường phía trên một kết nối được mã hóa bởi SSL hoặc TLS: tất cả nội dung trong thông điệp HTTP đều được mã hóa, kể cả tiêu đề của gói tin. Độ tin cậy của HTTPS là rất cao vì ngoại trừ tấn công CCA (sẽ nói sau) còn thì kẻ tấn công nếu nắm bắt được thông điệp cũng sẽ chỉ biết được địa chỉ IP đến và đi của thông điệp (mà điều này thì họ đã biết rồi) còn ngoài ra không thể hiểu gì.

Cài đặt máy chủ

Để chuẩn bị cho một website tiếp nhận liên kết HTTPS, người quản trị cần tạo một khóa công khai cho máy chủ web. Chứng thư cấp cho khóa này phải được ký xác nhận bởi một CA đáng tin cậy đối với trình duyệt để được tiếp nhận. CA cần chứng nhận rằng người mang chứng thư đúng thực là thực thể mà người đó đăng ký. Các trình duyệt thường được phân phối kèm theo những chứng thư được ký bởi đa số các CA do đó có thể thẩm định được các chứng thư do những CA đó ký xác nhận.

Tiếp nhận chứng thư

Các chứng thư có thể được cấp miễn phí bởi một số CA, một số khác yêu cầu nộp lệ phí duy trì không lớn (năm 2010 là từ 13USD cho đến khoảng 1,500USD mỗi năm). Các tổ chức lớn, có uy tín cũng có thể cho lưu hành chứng thư do CA của chính tổ chức mình phát hành, đặc biệt trong trường hợp họ thiết kế lấy trình duyệt để truy cập các website của họ (chẳng hạn các mạng Intranet của các công ty, của các Đại học lớn). Các tổ chức này cũng có thể gắn thêm bản sao chứng thư tự tạo của họ vào các chứng thư đáng tin cậy được phân phối cùng với trình duyệt. Cũng có thể có những tổ chức chứng thực lẫn nhau được gọi là CACert.

Tích hợp trình duyệt

Hầu hết các trình duyệt khi nhận được một chứng thư không có giá trị đều đưa ra một cảnh báo. Các trình duyệt loại cũ hơn thì mỗi khi kết nối với một website có chứng thư không hợp lệ thường đưa ra một hộp thoại cho người sử dụng và hỏi họ có muốn tiếp tục kết nối hay không. Các trình duyệt mới hơn thì đưa ra cảnh báo hiện trên toàn bộ cửa sổ. Các trình duyệt mới nhất gần đây còn có thể trình thông tin về sự an toàn của từng website ngay trong thanh địa chỉ.

Sử dụng để quản lý đăng nhập

Hệ thống cũng có thể sử dụng để nhận dạng phía khách nhằm hạn chế chỉ cho những người sử dụng được cấp phép mới có thể truy cập máy chủ web. Muốn làm điều này, người quản trị website sẽ tạo cho mỗi người sử dụng một chứng thư, chứng thư đó được tải vào trình duyệt của nó. Thông thường chứng thư đó gồm tên và địa chỉ E-mail của người dùng được cấp phép và được máy chủ kiểm tra tự động để thẩm định căn cước của người dùng ngay mỗi lần kết nối lại, không cần đến nhập mật khẩu.

Trường hợp bị lộ khóa riêng

Một chứng thư có thể bị hủy trước khi hết hạn, chẳng hạn vì lý do là khóa riêng ứng với nó đã bị lộ. Các trình duyệt mới gần đây như Google Chrome, Firefox, Opera và Internet Explorer trên Windows Vista được bổ sung thêm giao thức trạng thái chứng thư trực tuyến OCSP (*Online Certificate Status Protocol*) để thẩm định điều đó. Trình duyệt sẽ gửi số serial của chứng thư cho CA hay cho đại diện của CA thông qua OCSP và CA trả lời ngay là chứng thư đã bị hủy hay chưa.

Một số điểm hạn chế

SSL không ngăn chặn được toàn bộ một website sử dụng một "đường lách" (*crawler*) và đôi khi có thể đoán ra được địa chỉ URL của nguồn đã mã hóa khi chỉ biết kích thước của các lệnh request và response yêu cầu/trả lời. Điều này làm cho kẻ tấn công dễ tiến hành thám mã.

Do bởi giao thức SSL hoạt động bên dưới HTTP và không hề biết gì về các giao thức ở các tầng trên cho nên các máy chủ SSL chỉ có thể trình ra được một chứng thư cho mỗi bộ địa chỉ IP/Cổng. Điều này có nghĩa là trong hầu hết các trường hợp, việc sử dụng cách đặt tên để tạo hosting ảo là không khả thi đối với HTTPS. Có một giải pháp cho điều này là tích hợp một phần mềm gọi là Chỉ thị tên máy chủ SNI (*Server Name Indication*). SNI sẽ gửi tên của host đến máy chủ trước khi mã hóa kết nối. Tuy nhiên chỉ có các trình duyệt mới từ Firefox-2, Opera-8, Safari 2.1, Google Chrome 6 và Internet Explorer 7 trên Windows Vista mới được hỗ trợ SNI, còn các browser cũ thì không tương thích.

6.3.2. S-HTTP

Bạn đừng lẫn lộn HTTPS với giao thức S-HTTP trong họ giao thức RFC 2660.

HTTP an toàn S-HTTP (*Secure HTTP*) là một giao thức truyền thông hướng thông điệp có bảo mật được sử dụng kết hợp với HTTP. S-HTTP được thiết kế nhằm cùng tồn tại với mô hình truyền thông điệp của HTTP và có thể tích hợp dễ dàng vào các ứng dụng của HTTP. Cần chú ý rằng: S-HTTP mã hóa từng thông điệp riêng lẻ trong khi HTTPS mã hóa toàn bộ một kênh truyền thông. Vì vậy S-HTTP không thể dùng để bảo vệ an toàn mạng riêng ảo VPN (*Virtual Private Network*) nhưng HTTPS thì lại được.

S-HTTP cung cấp hàng loạt cơ chế an ninh cho phía máy khách và phía máy chủ của HTTP, những cơ chế này cung cấp các dạng dịch vụ an ninh phù hợp với nhiều mục đích sử dụng rộng rãi cho WWW. S-HTTP cung cấp những khả năng hoàn toàn đối xứng và bình đẳng cho phía máy khách và phía máy chủ mà vẫn giữ nguyên mô hình giao tiếp và các đặc tính của HTTP.

Nhiều dạng tiêu chuẩn mã hóa thông điệp được tích hợp vào phía máy khách và máy chủ S-HTTP. S-HTTP hỗ trợ các tương tác trong hàng loạt hoạt động tương thích với HTTP. S-HTTP không đòi hỏi chứng thư khóa công khai của phía máy khách vì nó chỉ hoạt động với hệ thống khóa đối xứng. Điều này rất có ý nghĩa vì thường có thể xuất hiện những giao dịch, thanh toán đột xuất không thể đòi hỏi người dùng cá nhân phải có sẵn một khóa công khai được thiết lập. Tuy là S-HTTP có ưu thế là có thể thiết lập hạ tầng cơ sở chứng thực khắp nơi nhưng để triển khai sử dụng nó thì lại không cần đến điều ấy.

S-HTTP hỗ trợ các giao dịch an toàn từ đầu đến cuối. Khách có thể "thoạt tiên" bắt đầu một giao dịch bí mật (điển hình là dùng những thông tin trong phần tiêu đề của thông điệp), điều đó có thể dùng để hỗ trợ việc mã hóa các mẫu phải điền chẳng hạn. Với S-HTTP thì không có dữ liệu nhạy cảm nào phải gửi đi dưới dạng tường minh trên mạng.

S-HTTP cung cấp những thuật toán, những phương thức và tham số mã hóa hoàn toàn mềm dẻo. Việc thương lượng để chọn lựa cho phép phía máy khách và phía máy chủ thỏa thuận về các thuật toán mã hóa cho phương thức giao dịch (RSA thay bởi DSA để ký, DES thay bởi RC2 để mã hóa v.v.) và tuyển chọn chứng thư.

S-HTTP được thực hiện nhằm tránh khỏi quá phụ thuộc vào một mô hình tin cậy riêng biệt nào đó dù rằng những người thiết kế ra nó thừa nhận giá trị và tạo điều kiện để dễ dàng thực hiện mô hình hệ thống tin cậy theo tôn ti từ gốc và cũng chấp nhận là có thể có nhiều chứng thực khóa công khai.

S-HTTP khác với *Digest-Authentication* ở chỗ là D-A có hỗ trợ cho khả năng sử dụng mã hóa khóa công khai và chữ ký số đồng thời cũng đảm bảo tính bí mật riêng tư.

6.3.3. FTPS

Giao thức truyền tệp có bảo mật (FTPS)

Giao thức truyền tệp có bảo mật FTPS (File Transfer Protocol Secure) là sự bổ sung kết hợp sự hỗ trợ của các giao thức bảo mật SSL hay TLS vào giao thức truyền tệp FTP. Thiết kế và hoạt động của FTPS cũng tương tự như HTTPS.

FTP được soạn thảo từ 1971 để sử dụng cho công tác trao đổi nghiên cứu khoa học trên liên mạng ARPANET. Vào thời đó việc truy cập vào ARPANET được hạn chế cho một số mạng quân sự và một vài trường đại học và chỉ có một cộng đồng người sử dụng rất hẹp mới có thể làm việc mà không yêu cầu tính bí mật hoặc riêng tư cho dữ liệu trong giao thức.

ARPANET phân rã một bộ phận thành liên mạng NSFnet, mạng này về sau trở thành Internet với số người sử dụng truy cập vào máy chủ thông qua những con đường truyền thông dài trên Internet tăng

lên rất nhiều cho nên cơ hội cho những kẻ "đọc trộm" những dữ liệu trao đổi cũng rất lớn.

Năm 1994, Công ty Netscape tung ra bộ giao thức SSL để bảo vệ cho việc truyền thông trên Internet chống lại sự đọc trộm thông tin. SSL được sử dụng kèm theo với HTTP để tạo ra giao thức truyền thông có bảo mật HTTPS và đến năm 1996 với bản phác thảo RFC (*Request for Comments*) giao thức SSL cũng bắt đầu được sử dụng kèm với giao thức truyền tệp FTP. Sau đó không lâu một cổng thông tin của IANA đã được đăng ký chính thức, tuy nhiên cũng phải đến năm 2005 thì RFC mới chính thức hoàn thành.

Các phương pháp gọi chức năng bảo vệ

Có hai phương pháp khác nhau được phát triển để sử dụng cho việc gọi chức năng bảo vệ an ninh phía máy chủ cho các máy khách sử dụng FTP: Phương pháp tường minh/phương pháp hiện (*Explicit*) và phương pháp ngầm/ẩn (*Implicit*). Phương pháp hiện là một sự bổ sung tương thích qua đó FTPS thông báo cho người sử dụng có thể gọi chức năng bảo vệ an ninh với một máy chủ (có bảo vệ FTPS) mà không gây ảnh hưởng đến hoạt động của FTP đối với những khách sử dụng không gọi đến FTPS. Phương pháp ẩn đòi hỏi mọi khách sử dụng máy chủ FTPS đều phải được cảnh báo là trong phiên giao dịch đó SSL đang được sử dụng và như vậy sẽ không tương thích với những khách không gọi đến FTPS.

Phương pháp tường minh

Trong phương pháp tường minh, cũng được gọi là **FTPES**, một khách sử dụng FTPS phải "nêu rõ ràng" yêu cầu sự bảo vệ an ninh từ phía một máy chủ FTPS và ngay tiếp sau đó là trao đổi thỏa thuận một khóa mã. Nếu phía khách không yêu cầu an ninh thì phía chủ có thể: hoặc là cho phía khách tiếp tục làm việc với chế độ không an toàn, hoặc là từ chối hay giới hạn việc kết nối.

Cơ chế để thương lượng về cách nhận dạng và bảo vệ an ninh với FTP được tăng cường bằng giao thức RFC 2228, bao gồm cả một lệnh FTP mới là **AUTH**. Trong khi RFC đó không quy định rõ ràng một cơ chế an ninh nào được yêu cầu (nghĩa là SSL hay TLS) nhưng lại đòi hỏi khách sử dụng FTPS phải trao đổi với máy chủ FTPS về một cơ chế an ninh mà cả đôi bên đều biết.

Nếu phía khách FTPS đưa ra cho phía máy chủ FTPS một cơ chế an ninh mà phía máy chủ không biết thì máy chủ FTPS sẽ trả lời với lệnh AUTH là có sai lầm mã số 504 (không được hỗ trợ) (*Error Code 504*). Phía khách có thể xác định xem là được hỗ trợ bởi cơ chế an ninh nào bằng cách yêu cầu máy chủ FTPS bằng lệnh FEAT, nhưng phía máy chủ không nhất thiết phải thông báo là nó sẽ hỗ trợ mức an ninh nào.

Các phương pháp chung thường gồm: AUTH TLS và AUTH SSL.

Trong phiên bản mới sau này RFC 4217, FTPS yêu cầu phía khách luôn luôn phải dùng phương pháp AUTH TLS để thương lượng. RFC cũng luôn khuyến cáo các phía máy chủ FTPS chấp nhận cơ chế AUTH TLS-C.

Phương pháp ẩn

Với các cấu trúc FTPS dạng ẩn, người ta không cho phép thương lượng. Phía khách ngay lập tức phải gửi đến phía máy chủ FTPS một thông điệp Hello TLS/SSL (*TLS/SSL ClientHello message*), nếu không nhận được thông điệp chào hỏi đó thì phía máy chủ ngắt ngay kết nối.

Để bảo đảm tương thích với những khách sử dụng FTP không dùng TLS/SSL, số này hiện nay vẫn có, FTPS dạng ẩn phải trông đợi được ở kênh quản lý FTPS và kênh 989/TCP về dữ liệu FTPS trên Cổng 990/TCP quen thuộc của IANA. Điều này cho phép các người quản trị bảo đảm được những dịch vụ tương thích hợp pháp trên kênh quản trị gốc 21/TCP FTP.

Nên nhớ rằng thương lượng dạng ẩn không được xác định trong RFC 4217. Do vậy đòi hỏi phải có một biện pháp thương lượng TLS/SSL cho FTP được tiến hành trước.

Hỗ trợ chung

FTPS hỗ trợ toàn phần cho các giao thức mã hóa TLS và SSL, bao gồm cả sử dụng của chứng thực thẩm định khóa công khai cho phía máy chủ lẫn sử dụng của chứng thư cấp phép cho phía khách. Nó cũng hỗ trợ các thuật toán mã hóa tương thích thông dụng như AES, RC4, Triple DES và các hàm băm SHA, MD5, MD4, và MD2.

Phạm vi ứng dụng

Trong phương thức ẩn, toàn bộ phiên FTPS đều được mã hóa. Phương thức hiện khác ở chỗ là phía khách có sự kiểm soát hoàn toàn về những vùng nào của liên kết cần được mã hóa. Có thể cho hoạt động hoặc ngừng hoạt động chức năng mã hóa cho kênh quản lý FTPS và của kênh dữ liệu FTPS bất cứ lúc nào. Điều hạn chế duy nhất là từ phía máy chủ vì nó có khả năng từ chối một số lệnh dựa trên chính sách mã hóa của máy chủ.

Kênh điều khiển an toàn

Phương thức Kênh điều khiển an toàn (*Secure Command Channel*) có thể đưa vào bằng cách phát xuất những lệnh AUTH TLS hay AUTH SSL. Sau mỗi lần như vậy, mọi truyền thông kênh dữ liệu giữa phía khách FTPS và phía máy chủ đều giả thiết là đã được mã hóa. Nói chung được khuyến cáo là nên nhập vào một trạng thái được mã hóa như vậy trước khi tiến hành nhận dạng và cấp phép cho người sử dụng, điều này nhằm chống kẻ thứ ba nghe trộm tên và mật khẩu của người sử dụng.

Kênh dữ liệu an toàn

Kênh dữ liệu an toàn có thể đưa vào thông qua việc phát xuất lệnh **PROT**. Điều này mặc định là *không được phép* khi đã có lệnh

AUTH TLS phát xuất trước đó. Sau mỗi lần như vậy, mọi truyền thông kênh dữ liệu giữa phía khách FPTS và phía máy chủ đều giả thiết là đã được mã hóa. Phía khách FTPS có thể thoát khỏi kiểu kênh dữ liệu an toàn bất cứ lúc nào bằng cách phát xuất một lệnh Xóa kênh dữ liệu CDC (*Clear Data Channel*).

Lý do để ngừng chức năng mã hóa

Khi thực hiện truyền thông tin dưới những tình huống như sau đây thì sử dụng việc mã hóa kênh dữ liệu có thể không lợi:

- Các tệp được truyền có nội dung bình thường, không có gì nhạy cảm, không cần phải mã hóa.
- Các tệp được truyền đã mã hóa từ trước trong tệp
- Mã hóa TLS hay SSL không đạt yêu cầu bảo mật. Điều này thường xảy ra đối với phần mềm phía khách và phía chủ FTPS đời cũ, bị giới hạn ở giao thức SSL 40 bit do luật cấm xuất khẩu phần mềm mã hóa cao cấp của Hoa Kỳ trước đây.

Trong những tình huống như sau thì sử dụng việc mã hóa kênh quản lý cũng có thể không có lợi:

- Sử dụng FTPS khi mà máy khách hoặc máy chủ đặt phía sau một tường lửa mạng hoặc một thiết bị thay đổi địa chỉ mạng NAT (*Network Address Translation*).
- Có những khách sử dụng FTP vô danh sử dụng lặp lại AUTH hay CCC/CDC cùng trong một phiên giao dịch. Hành vi như vậy có thể dùng cho một tấn công từ chối dịch vụ DOS (*Denial of Service*) vì rằng cứ mỗi lần như vậy thì một phiên TLS/SSL lại phải được tạo ra làm mất thời gian xử lý của máy chủ.

Chứng thư SSL

Giống như HTTPS (nhưng khác với SFTP), các máy chủ FTPS có thể cung cấp chứng thực khóa công khai.

Các chứng thực đó có thể được tạo ra bằng cách sử dụng những công cụ của Unix như là *ssl-ca* của **OpenSSL**.

Chứng thực đó phải được một cơ quan chứng thực điện tử (CA) ký nếu không thì phía khách FTPS sẽ thấy mình được cảnh báo là chứng thư không hợp lệ.

Không tương thích với tường lửa

Do bởi FTP sử dụng một cổng thứ cấp động (cho các kênh dữ liệu) nên nhiều tường lửa được thiết kế để luôn dò xét các thông điệp trong giao thức FTP nhằm xác định xem những liên kết dữ liệu thứ cấp nào cần được cấp phép. Thế nhưng, nếu kết nối quản lý FTP được dùng TLS/SSL để mã hóa thì tường lửa không thể xác định được số hiệu Cổng của các dữ liệu trao đổi giữa phía khách và phía chủ FTP.

Vì thế trong nhiều mạng máy tính có tường lửa, khi triển khai các tệp không mã hóa thì được nhưng với các tệp mã hóa thì lại không triển khai được.

Vấn đề này có thể giải quyết bằng cách là chỉ sử dụng một số ít Cổng cho dữ liệu và ta sẽ định dạng cho tường lửa chấp nhận các cổng đó.

Không nên nhầm lẫn FTPS với SFTP (SSH File Transfer Protocol: Giao thức truyền tệp bao vỏ sò), một hệ thống con của SSH dùng để truyền các tệp tin lớn.

Có rất nhiều cơ chế để truyền tệp sử dụng giao thức SSH:

- Secure copy (SCP), được phát triển từ giao thức RCP trên SSH

- SSH File Transfer Protocol (SFTP), một giao thức truyền tệp có bảo mật bằng SSH thay thế cho ETP.

- FTP trên SSH (A.K.A. FISSH) đưa ra từ năm 1998, được phát triển từ các lệnh Unix shell trên SSH.

6.4. SSH

6.4.1. Giao thức vỏ sò bảo mật (SSH)

Giao thức vỏ sò bảo mật SSH (Secure Shell Protocol) là một giao thức mạng an toàn để kiểm tra và bảo vệ việc truy cập từ xa trong dịch vụ TELNET và cũng có thể mã hóa để bảo mật dữ liệu trong dịch vụ truyền các tệp tin điện tử lớn (FTP) trong môi trường không tin cậy chẳng hạn như môi trường Internet.

SSH cho phép trao đổi dữ liệu giữa 2 thiết bị mạng thông qua một kênh tin cậy. Hai phiên bản chính của SSH là SSH1 hay SSH-1 và SSH2 hay SSH-2.

Vị trí của SSH trong chuỗi giao thức Internet

Chuỗi giao thức Internet
Tầng ứng dụng
BGP · DHCP · DNS · FTP · HTTP · IMAP · IRC · LDAP · MGCP · NNTP · NTP · POP · RIP · RPC · RTP · SIP · SMTP · SNMP · **SSH** · Telnet · TLS/SSL · XMPP · …..
Tầng giao vận
TCP · UDP · DCCP · SCTP · RSVP · ECN ·
Tầng mạng
IP (IPv4, IPv6) · ICMP · ICMPv6 · IGMP · IPsec ·
Tầng liên kết dữ liệu
ARP/InARP · NDP · OSPF · Tunnels (L2TP) · PPP · Media Access Control (Ethernet, DSL, ISDN, FDDI) · (more)
v. v…

SSH sử dụng mật mã khóa công khai để nhận dạng một máy tính ở xa đồng thời cũng cho phép máy tính ở xa nhận dạng được người đang kết nối sử dụng nếu cần thiết. Cổng tiêu chuẩn TCP 22 SSH quy định sử dụng để liên kết với các máy chủ SSH.

6.4.2. Phiên bản 1.x

Năm 1995, Tatu Ylönen một nghiên cứu viên tại Đại học Công nghiệp Helsinki, Phần Lan (*Helsinki University of Technology*) đã thiết kế phiên bản đầu tiên của giao thức ngày nay gọi là SSH-1 ngay sau khi cảm nhận có nguy cơ tấn công đánh cắp mật khẩu trong mạng máy tính của trường đại học. Mục đích của giao thức đó là để tăng cường bảo vệ cho các giao thức TELNET, rlogin và rsh đang được sử dụng trong mạng. Công trình của Ylönen được cho phép sử dụng tự do từ tháng 7 năm 1995 và nhanh chóng trở nên phổ biến. Đến cuối 1995 số người sử dụng cơ sở SSH đã lên đến hơn 20.000 người ở trên 50 quốc gia.

Đến tháng 12 năm 1995 Ylönen thành lập tổ chức An ninh truyền thông SSH (*SSH Communications Security*) để tiếp thị và phát triển SSH. Phiên bản gốc của SSH sử dụng nhiều bộ phận của những phần mềm miễn phí như là GNU libgmp nhưng về sau đó các phần mềm do Công ty an ninh truyền thông SSH dần phát triển thành những phần mềm có bản quyền. Đến năm 2000 người ta ước lượng có khoảng hơn 2 triệu người sử dụng SSH.

Mã độc tấn công

Vào năm 1998 một mã độc được mô tả trong phiên bản SSH 1.5, mã độc này có thể chèn những nội dung bất hợp pháp vào các dòng dữ liệu mã hóa do bởi phiên bản này có sử dụng CRC-32 vốn không đủ khả năng bảo vệ toàn vẹn dữ liệu. Sau đó mọi phiên bản sau đều được tích hợp một phần mềm diệt mã độc gọi là bộ phát hiện tấn công của SSH. Tháng giêng năm 2001 người ta phát hiện một mã độc có khả năng cho phép kẻ tấn công làm thay đổi khối (*block*) cuối

cùng của một phiên mã hóa IDEA. Cũng trong tháng đó người ta lại phát hiện thêm một mã độc có khả năng cho phép một máy chủ mạo danh có thể chuyển tiếp việc nhận dạng khách sử dụng sang một máy chủ khác.

Phiên bản 1.99

Tháng giêng năm 2001, sau khi phiên bản 2.1 được xây dựng xong, RFC 4253 đã quy định rằng một máy chủ SSH hỗ trợ cả phiên bản 2.0 và cả các phiên bản trước đó được gọi là phiên bản 1.99. Đấy không phải là một phiên bản hiện tại mà chỉ là một cách để nhận rõ khả năng tái lập của nó (*backward compatibility*).

OpenSSH và OSSH

Năm 1999, các nhà phát triển phần mềm muốn có một phiên bản sử dụng tự do nên đã quay lại với phiên bản 1.2.12 của chương trình SSH đầu tiên, đấy là phiên bản cuối cùng được phát hành dưới dạng mã nguồn mở. OSSH của Björn Grönvall được phát triển trên cơ sở đó. Không lâu sau đó các nhà phát triển phần mềm OpenBSD đã phát triển công trình của Grönvall' và tạo ra Open SSH ra đời đồng thời với phiên bản 2.6 của OpenBSD. Từ phiên bản đó, OpenSSH đã được mang sử dụng cho nhiều hệ điều hành khác.

Đến năm 2005 thì SSH vẫn là phiên bản phổ biến duy nhất của SSH được sử dụng mặc định trong rất nhiều hệ điều hành. Mãi đến nay (2011) OpenSSH vẫn còn được dùng và hiện đang hỗ trợ cả các phiên bản 1.x và 2.0.

6.4.3. Phiên bản 2.x. "Secsh"

Là tên gọi chính thức của Lực lượng công tác kỹ thuật Internet IETF (*Internet Engineering Task Force*) đặt cho bộ phận của IETF chịu trách nhiệm phát triển phiên bản 2 của giao thức SSH. Năm 2006, một phiên bản được duyệt lại của giao thức là SSH-2 được thừa nhận là phiên bản tiêu chuẩn. Phiên bản này không tương thích với SSH-1

và vượt trội hơn SSH-1 cả về độ bảo mật cũng như về phạm vi nội dung phát triển. Chẳng hạn như về độ bảo mật là do việc sử dụng sơ đồ trao đổi khóa Diffie-Helman và về bảo vệ toàn vẹn thông tin là do sử dụng các mã nhận dạng thông điệp. Một nội dung mới nữa của SSH-2 là khả năng chạy được một số phiên *shell* bất kỳ chỉ trên một kết nối đơn SSH.

Mã độc

Tháng 11 năm 2008 người ta phát hiện một mã độc xâm nhập mọi phiên bản của SSH, mã độc đó cho phép tái hiện 32 bit của plaintext từ một khối của ciphertext đã được mã hóa bằng cách sử dụng thuật toán mã hóa đối xứng được mặc định là tiêu chuẩn CBC.

Các phiên bản tiêu chuẩn của SSH

Tổ chức IETF đề xuất danh mục các phiên bản sau đây của SSH được xem là tiêu chuẩn sử dụng trên Internet.

- RFC 4250, The Secure Shell (SSH) Protocol Assigned Numbers.
- RFC 4251, The Secure Shell (SSH) Protocol Architecture
- RFC 4252, The Secure Shell (SSH) Authentication Protocol
- RFC 4253, The Secure Shell (SSH) Transport Layer Protocol
- RFC 4254, The Secure Shell (SSH) Connection Protocol
- RFC 4255, Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints
- RFC 4256, Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)
- RFC 4335, The Secure Shell (SSH) Session Channel Break Extension
- RFC 4344, The Secure Shell (SSH) Transport Layer Encryption Modes

- RFC 4345, Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer Protocol

Sau đó có thêm những phiên bản nâng cấp:

- RFC 4419, Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol (March 2006)
- RFC 4432, RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol (March 2006)
- RFC 4462, Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol (May 2006)
- RFC 4716, The Secure Shell (SSH) Public Key File Format (November 2006)
- RFC 5656, Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer (December 2009)

Do bởi SSH-1 có những lỗ hổng cố hữu trong thiết kế nên có thể bị tấn công của người đứng giữa. Nên ngày nay, người ta xem như đã lỗi thời, không còn sử dụng nữa. Các phần mềm phía máy chủ và phía máy khách hiện đại đều được hỗ trợ sử dụng SSH-2.

Tuy nhiên trong mọi phiên bản của SSH điều tối quan trọng là việc thẩm tra khóa công khai của "người lạ" trước khi chấp nhận rằng đấy là những khóa hợp lệ. Việc chấp nhận một khóa công khai của kẻ tấn công giấu mặt làm khóa hợp lệ có hệ quả nguy hiểm là làm lộ các mật khẩu chuyển giao trong hệ thống và tạo điều kiện cho sự tấn công của người đứng giữa.

6.5. THANH TOÁN ĐIỆN TỬ AN TOÀN

6.5.1. SET

Thanh toán điện tử an toàn SET (Secure Electronic Transaction) là một giao thức chuẩn để đảm bảo an toàn thanh toán cho các thẻ

tín dụng trên một mạng truyền thông không tin cậy, nhất là trên Internet.

Bản thân SET không phải là một hệ thống thanh toán mà thực ra là một tập hợp giao thức và thủ tục cho phép người dùng có thể thực hiện cơ chế sẵn có của một hệ thống thanh toán thẻ một cách an toàn trong môi trường mở.

SET được phát triển bởi SETco, một công ty an ninh mạng do VISA và MasterCard chỉ đạo, kể từ 1996 và sau đó một số công ty khác như GTE, IBM, Microsoft, Netscape, RSA và VeriSign cũng tham gia. SET cơ bản dựa trên chuẩn X.509 với một số tiêu chuẩn mở rộng. Phiên bản đầu tiên hoàn thành vào tháng 5 năm 1997 và bản dùng thử lần đầu tiên thử nghiệm vào tháng 7 năm 1998.

SET cho phép các bên đối tác nhận dạng ra nhau (thông tin nhận dạng đã mã hóa) và sau đó trao đổi thông tin một cách an toàn. SET dùng một thuật toán cho phép người bán hàng thay thế một chứng thư cho một số của thẻ tín dụng của người sử dụng.

Bản thân người bán hàng không bao giờ cần biết đến số của thẻ tín dụng mà người dùng (người mua) gửi đến, mà vẫn kiểm tra được việc thanh toán trả tiền mặt khác bảo vệ được cho chủ thẻ và nhà phát hành thẻ khỏi bị lừa đảo.

Ngày nay SET thực tế đã trở thành giao thức tiêu chuẩn cho việc thanh toán trên Internet giao dịch giữa người bán hàng, người mua và các công ty phát hành thẻ. Một hệ thống SET bao gồm các thành viên sau đây:

- Chủ thẻ
- Người bán hàng
- Nhà phát hành thẻ
- Nơi chấp nhận thẻ

- Cổng thanh toán
- Tổ chức chứng thực điện tử

Hệ thống giao diện của SET có 3 thành phần:

- Giao diện ví điện tử: được cài đặt trong thẻ/máy tính của người trả tiền (người mua).
- Giao diện ở máy tính/máy đọc thẻ của người nhận tiền (người bán).
- Giao diện tại máy chủ của ngân hàng phát hành thẻ liên kết với máy chủ ngân hàng có tài khoản của người nhận tiền.

6.5.2. Hoạt động thanh toán

Quy trình thanh toán diễn ra như sau:

1. Khách hàng yêu cầu và nhận được một tài khoản thẻ tín dụng từ một ngân hàng có hỗ trợ thanh toán điện tử và SET.
2. Khách hàng nhận một chứng thư số X509v3 do ngân hàng ký xác nhận.
3. Người bán cũng có chứng thư số của họ
4. Khách hàng lập phiếu đặt hàng
5. Người bán gửi một bản sao chứng thư của mình để cho khách hàng có thể kiểm tra xác minh rằng đấy là một của hàng hợp lệ
6. Gửi phiếu đặt hàng và lệnh chi trả
7. Người bán yêu cầu kiểm tra sự cho phép chi trả
8. Người bán xác nhận phiếu đặt hàng
9. Người bán gửi hàng hóa hay dịch vụ đến cho người mua
10. Người bán yêu cầu chi trả

Trong tiêu dùng trực tiếp khi người mua đưa thẻ để trả tiền, người bán cắm vào máy để máy đọc và ghi lại toàn bộ thông tin đã mã hóa gửi đến cho ngân hàng cấp thẻ. Tại đấy các thông tin được giải mã, ngân hàng nhận diện (tài khoản) của người trả tiền và của người nhận tiền, nếu đúng sẽ thông báo chấp nhận thanh toán và bộ phận kế toán thực hiện việc chuyển khoản từ tài khoản người trả tiền đến tài khoản người nhận tiền. Khi thực hiện xong (hoặc chấp nhận thực hiện) bộ phận kế toán tại ngân hàng thông báo cho cả hai bên người trả tiền và người nhận tiền là giao dịch đã hoàn thành.

Nói chung hiện nay các ngân hàng đang thực hiện giao dịch thanh toán thẻ tín dụng qua mạng đều tin tưởng vào hệ thống SET vì hệ thống này đảm bảo việc nhận dạng chính xác các đối tác trả tiền và nhận tiền đồng thời không cho phép người nhận tiền giải mã để nắm được thông tin trong thẻ của người trả tiền, điều này giảm bớt nguy cơ bị trộm thông tin thẻ (*pharming*).

6.5.3. Chữ ký song hành

Sáng tạo độc dáo của SET là phương pháp sử dụng chữ ký song hành (*Dual signature*). Chữ ký song hành cũng có chức năng tương tự như chữ ký điện tử là xác nhận người phát hành thông tin và sự toàn vẹn thông tin.

Muốn vậy SET tổ chức kết nối để so sánh hai bản tin được gửi cho hai hòm thư khác nhau. Giả sử người mua chỉ gửi thông tin mua hàng OI (*Order Information*) cho người bán hàng và thông tin trả tiền PI (*Payment Information*) cho ngân hàng, như vậy người bán không biết gì về thông tin tài khoản của người mua, ngân hàng cũng không cần biết về thông tin hàng hóa. Giá trị băm của thông tin mua hàng và của thông tin trả tiền được mã hóa bởi các khóa riêng (khác nhau) của người mua hàng thành một cặp chữ ký, cặp chữ ký đó được gắn cả vào từng thông điệp OI và PI để gửi cho cả người bán và ngân hàng. Người bán giải mã giá trị băm của OI để kiểm tra và lưu giá trị băm của PI làm chứng từ đối chiếu nếu sau này cần thiết,

nhưng không biết nội dung của PI. Ngược lại ngân hàng giải mã được giá trị băm của PI để kiểm tra nhưng lại không thể biết nội dung của OI.

SET là một hệ thống thanh toán đảm bảo được các yêu cầu: xác nhận được các đối tác tham gia giao dịch, không thể chối bỏ, thông tin thanh toán minh bạch và được bảo mật an toàn, thực hiện thanh toán nhanh.

Tuy nhiên vì các chi tiết giao dịch và đối tác đều được lưu ít nhất là trên bộ phận kế toán ngân hàng cho nên việc thanh toán không đảm bảo được bí mật, riêng tư cho người mua và người bán.

6.6. IPsec

Giao thức Internet an toàn IPsec (Internet Protocol Security) là sự nối tiếp của giao thức an ninh tầng mạng của mô hình chuẩn ISO *NLSP* (*Network Layer Security Protocol*). NLSP lại dựa trên cơ sở của giao thức SP3 được NIST công bố nhưng lại được thiết kế do dự án An ninh Dữ liệu Hệ thống Mạng của NSA (National Security Agency: Cơ quan an ninh quốc gia) của Hoa Kỳ.

Các tổ chức, công ty… khi trao đổi dữ liệu giữa các máy tính từ Hội sở chính đến các chi nhánh đều có yêu cầu phải bảo mật dữ liệu của mình không để lọt ra ngoài. Sử dụng đường truyền kết nối trực tiếp (*leased line*) là một biện pháp hữu hiệu để thực hiện điều đó, tuy nhiên giá thành của biện pháp này quá cao và việc truyền thông lại không được linh hoạt như truyền thông qua Internet. Hiện nay phương án thương mại được người ta thích lựa chọn là việc tạo một mạng riêng ảo VPN (*Virtual Private Network*) dựa trên cơ sở của giao thức Internet an toàn.

Ngày nay, truyền thông điện tử B2B (*Business to Business*) đã trở thành một nhu cầu sống còn đối với các doanh nghiệp. Chẳng hạn, một doanh nghiệp ký hợp đồng bao thầu cung cấp cho một khách sạn lớn thường xuyên phải truy vấn vào cơ sở dữ liệu của

khách sạn để kịp thời cung cấp vật tư, hàng hóa cho khách sạn: điều này rất cần thiết và có lợi cho cả khách sạn lẫn nhà cung cấp.

Thế nhưng về phía khách sạn, có nhiều thông tin, dữ liệu của mạng nội bộ cần được bảo mật chẳng hạn như danh sách khách hàng, các dữ liệu kế toán v.v. vì vậy nhu cầu bảo mật từng bộ phận dữ liệu trong một mạng nội bộ, phân quyền cho những lớp người sử dụng khác nhau khi truy cập vào một mạng máy tính nhất là thông qua Internet, là một vấn đề có tầm quan trọng rất lớn đối với việc kinh doanh của doanh nghiệp.

6.6.1. Khả năng xác thực

IPsec cung cấp khả năng xác thực (*authentication*), bí mật, toàn vẹn thông tin, quản lý truy cập, chống tấn công phân tích luồng dữ liệu, nói chung là có khả năng nhận dạng được mọi gói dữ liệu vào và mã hóa mọi gói dữ liệu ra của một mạng cục bộ.

IPsec là một chuỗi giao thức nhằm bảo vệ các truyền thông qua giao thức Internet (IP) bằng cách xác thực và mã hóa mỗi gói tin IP của từng phiên giao dịch. IPsec cũng bao gồm cả những giao thức nhằm thiết lập sự xác thực lẫn nhau giữa các đối tác khi khởi đầu một phiên và sự thương lượng để thỏa thuận các khóa mật mã dùng cho phiên giao dịch đó.

IPsec là một sơ đồ bảo vệ an ninh các thiết bị đầu cuối hoạt động ở tầng Internet của chuỗi giao thức Internet. Nó có thể sử dụng để bảo vệ luồng dữ liệu giữa hai máy khách *(host-to-host)* giữa hai mạng *(network-to-network)* hoặc giữa một mạng và một máy khách (*network-to-host*)

Một số hệ thống an ninh khác được sử dụng rộng rãi như SSL, TLP, SSH hoạt động ở những tầng trên của mô hình TCP/IP. IPsec hoạt động ở phía dưới tầng ứng dụng và là trong suốt (*transparent*) đối với mọi người sử dụng. Vì vậy IPsec bảo vệ cho mọi truyền thông

ứng dụng qua một mạng IP: E-mail, trình duyệt web, các file truyền đi ... và nói chung là mọi truyền thông điện tử giữa một máy tính với mọi máy tính khác cũng có cài đặt IPsec. Các ứng dụng không cần phải thiết kế đặc biệt để sử dụng được IPsec, trong khi muốn sử dụng TLS/SSL, người ta phải thiết kế thành một ứng dụng riêng để bảo vệ các giao thức ứng dụng.

IPsec được IETF tạo nên một dãy tư liệu Yêu cầu bình luận (*Request for Comment documents*) gửi đến các thành phần khác nhau trong mạng và vì thế có tên gọi của giao thức là IPsec.

Dãy IPsec là một **chuẩn mở** sử dụng các giao thức sau đây để thực hiện các hàm.

6.6.2. Tiêu đề xác thực (AH)

Một trong những thành phần của chuỗi giao thức IPsec protocol suite là Authentication Headers. AH đảm bảo sự toàn vẹn thông tin liên tục và kiểm tra địa chỉ nguồn của các gói tin IP. Ngoài ra nó còn bảo vệ chống kiểu tấn công lặp lại (replay attacks) bằng cách dùng kỹ thuật "cửa sổ trượt" và kỹ thuật "dập" tất cả các gói tin cũ.

Trong IPv4, AH bảo vệ các gói IP và mọi trường tiêu đề của một bản thông điệp chỉ trừ các trường thường có sự biến đổi. Các trường tiêu đề có biến đổi là: DSCP/TOS, ECN, Flags, Fragment Offset, TTL và Header Checksum.

Trong IPv6, AH tự bảo vệ ngay chính nó, bảo vệ tiêu đề mở rộng các mục tiêu đến (*Destination Options*) sau AH, và gói tin IP. Nó cũng bảo vệ cả tiêu đề IPv6 cố định và các tiêu đề mở rộng trước AH ngoại trừ các tiêu đề có thay đổi như DSCP. ECN, Flow Label và Hop Limit.

AH hoạt động trực tiếp trên đỉnh IP, sử dụng giao thức IP số hiệu 51.

Các sơ đồ gói AH sau đây chỉ rõ cách thức kiến tạo và minh họa một gói AH (Bảng 6.1):

Bảng 6.1

Định dạng Authentication Header																															
Offsets	$Octet_{16}$	0								1								2								3					
$Octet_{16}$	Bit_{10}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Next Header								Payload Len								Reserved															
4	32	Security Parameters Index (SPI)																															
8	64	Sequence Number																															
C	96	Integrity Check Value (ICV) …																															
…	…																																

Next Header (8 bit): Tiêu đề kế tiếp

Kiểu của tiêu đề kế tiếp, chỉ ra rằng giao thức tầng trên được bảo vệ như thế nào. Giá trị được lấy từ bảng liệt kê số hiệu của các giao thức IP.

Payload Len (8 bit)

Độ dài của tiêu đề xác thực (*Authentication Header*) tính theo đơn vị 4-octet trừ 2 (một giá trị của 0 là 8 octets, 1 là 12 octets, v.v.). Mặc dù kích thước được đo theo đơn vị 4-octet, độ dài của tiêu đề đó cần phải là một bội số của 8 octets nếu được mang bởi một gói tin IPv6. Điều này không cần thiết đối với các gói tin IPv4.

Reserved (16 bit): Dự trữ

Dự trữ sử dụng sau (mọi số 0 cho đến lúc đó).

Security Parameters Index (32 bit): Chỉ số các tham số an ninh

Một giá trị tùy chọn được sử dụng cùng với địa chỉ nguồn IP để nhận dạng tổ hợp an ninh (*security association*) của phía gửi thông điệp.

Sequence Number (32 bit): Số hiệu chuỗi

Một dãy đơn điệu tăng ngặt nhằm ngăn ngừa tấn công lặp lại.

Integrity Check Value (multiple of 32 bit): Giá trị kiểm tra tính toàn vẹn

Một giá trị có độ dài thay đổi, nó chứa các dãy để có thể triển khai ra trong một trường có biên 8-octet đối với IPv6 hoặc trường có biên 4-octet đối với IPv4.

6.6.3. Khối đóng gói an toàn

Khối đóng gói an toàn ESP (*Encapsulating Security Payloads*) cung cấp khả năng bảo mật, khả năng xác thực nguồn của dữ liệu,

kiểm tra tính toàn vẹn, dịch vụ chống tấn công lặp lại. ESP cũng là một thành phần trong dãy giao thức IPsec. Trong IPsec, ESP tạo ra chức năng xác thực nguồn, toàn vẹn, và bảo vệ bí mật riêng tư cho các gói tin. ESP cũng hỗ trợ các cấu hình "chỉ mã hóa" hoặc "chỉ giải mã" nhưng hành động mã hóa mà không có nhận dạng được khuyến cáo là không nên sử dụng vì kém an toàn.

Không giống như AH, ESP dùng trong chế độ "vận chuyển" (*Transport mode*) không cung cấp khả năng bảo vệ toàn vẹn và nhận dạng cho toàn bộ gói IP. Tuy nhiên trong kiểu "đường ống" (*Tunnel mode*) khi mà toàn bộ gói tin TP gốc được đóng gói lại và gắn một tiêu đề mới thêm vào thì ESP bảo vệ cho tất cả gói tin IP bên trong đó (kể cả tiêu đề bên trong) trong khi tiêu đề bên ngoài vẫn không được bảo vệ.

ESP hoạt động trên đỉnh của IP, sử dụng số hiệu IP là 50.

Các sơ đồ gói ESP packet sau đây chỉ rõ cách thức kiến tạo và minh họa một gói ESP (Bảng 6.2).

Bảng 6.2

Encapsulating Security Payload định dạng																															
Offsets	$Octet_{16}$	0								1								2								3					
$Octet_{16}$	Bit_{10}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Security Parameters Index (SPI)																													
4	32	Sequence Number																													
C	96	Payload data																													
…	…																														
…	…																														
…	…									Padding (0-255 octets)																					
…	…																	Pad length								Next Header					
…	…	Integrity Check Value (ICV) …																													
…	…																														

Security Parameters Index (32 bit): Chỉ số các tham số an ninh

Đây là một giá trị tùy ý chọn được sử dụng (cùng với địa chỉ nguồn IP) để nhận dạng *tổ hợp an ninh* của phía gửi tin.

Sequence Number (32 bit): Số hiệu chuỗi

Là một dãy số đơn điệu tăng (với mỗi gói tin gửi đi thì tăng thêm 1) nhằm chống kiểu tấn công lặp lại. Có một bộ đếm riêng cho mỗi tổ hợp an ninh.

Payload data (biến thiên): Dữ liệu đóng gói

Nội dung được bảo vệ của gói tin IP gốc, bao gồm cả mọi dữ liệu sử dụng để bảo vệ nội dung của nó (tức là một "Véc-tơ khởi đầu" của thuật toán mã hóa). Loại của nội dung được bảo vệ được chỉ rõ trong trường *tiêu đề kế tiếp.*

Padding (0-255 octets): Lớp đệm

Lớp đệm dùng cho mã hóa nhằm để mở rộng dữ liệu được đóng gói đạt đến kích thước phù hợp với một *khối mã hóa* và vừa với kích thước của trường kế tiếp.

Pad Length (8 bit): Độ dài đệm

Kích thước của lớp đệm tính theo đơn vị octet.

Next Header (8 bits): Tiêu đề kế tiếp

Kiểu của tiêu đề kế tiếp. Giá trị được lấy trong danh sách số hiệu của các giao thức IP.

Value (Bội số của 32 bit): Giá trị

Giá trị kiểm tra độ dài biến thiên. Nó có thể có một lớp đệm để cho trường đang xét phù hợp với một trường biên 8-octet đối với IPv6 hoặc 4-octet đối với IPv4.

6.6.4. Tổ hợp an ninh (SA)

Tổ hợp an ninh SA (Security associations): Cung cấp một gói thuật toán và dữ liệu sản sinh ra những tham số cần thiết để kích

hoạt các hoạt động của AH và ESP. *Internet Security Association và Key Management Protocol (ISAKMP)* tạo nên một khung cho hoạt động xác thực và trao đổi khóa với những bộ công cụ phổ biến hiện nay *Internet Key Exchange (IKE and IKEv2), Kerberized Internet Negotiation of Keys (KINK), hoặc IPSECKEY DNS records.*

Kiến trúc của IPsec sử dụng quan điểm về một "*tổ hợp an ninh*" làm cơ sở cho việc xây dựng các hàm an ninh vào trong IP. Một tổ hợp an ninh đơn giản chỉ là một gói gồm các thuật toán và các tham số (như là các khóa) sẽ được dùng để mã hóa và nhận dạng một luồng thông tin cụ thể theo một hướng. Do vậy, trong các lưu thông hai chiều thông thường, các luồng lưu thông được đảm bảo an ninh bằng một cặp tổ hợp an ninh.

Các tổ hợp an ninh được thiết lập bằng cách dùng Tổ hợp an ninh Internet và Giao thức quản lý khóa (ISAKMP). ISAKMP được trang bị bằng một cấu hình thủ công với những bí mật đã trao đổi trước như Trao đổi khóa Internet - *Internet Key Exchange (IKE and IKEv2)*, Thương lượng khóa Internet Kerberos - *Kerberized Internet Negotiation of Keys (KINK)*, và sử dụng IPSECKEY các bản ghi DNS.

Để quyết định dạng bảo vệ nào được cung cấp cho một gói tin sẽ gửi đi, IPsec sử dụng chỉ số tham số an ninh SPI (*Security Parameter Index*), một chỉ số cho cơ sở dữ liệu của tổ hợp an ninh SADB (*Security Association Database*), đồng thời với địa chỉ đích trong tiêu đề của gói tin. Một quy trình tương tự cũng được dùng để bảo vệ các gói tin đến, khi đó IPsec sẽ thu thập các khóa giải mã và xác thực từ cơ sở dữ liệu của tổ hợp an ninh.

Khi giao dịch với một nhóm nhiều đối tác, một tổ hợp an ninh được cung cấp cho cả nhóm và được sao gửi đến cho mọi người nhận tin trong nhóm. Có thể dùng các SPI để cấp cho các đối tác trong nhóm một số tổ hợp an ninh nhiều hơn và như vậy sẽ làm tăng mức độ an ninh trong nội bộ nhóm. Thật vậy, khi đó mỗi người gửi tin trong nhóm có thể có nhiều tổ hợp an ninh để nhận dạng đối tác

trong khi người nhận tin chỉ có thể biết là đã có một người nào đó biết được khóa và đã gửi tin cho mình.

6.6.5. Các chế độ hoạt động

IPsec có thể thực hiện theo chế độ "vận chuyển" từ máy khách đến máy khách đồng thời cũng có thể được thực hiện theo kiểu "đường ống" trong mạng máy tính.

Chế độ vận chuyển

Trong chế độ vận chuyển, thông thường chỉ có phần đóng gói (tức là dữ liệu được truyền đi) của gói tin IP là được mã hóa hay được nhận dạng. Tuyến đường vận chuyển là không thay đổi vì rằng tiêu đề của gói tin IP không hề bị thay đổi mà cũng không bị mã hóa, tuy nhiên, khi sử dụng tiêu đề xác thực thì địa chỉ IP không thể được phiên dịch vì như vậy sẽ ảnh hưởng đến giá trị băm. Các tầng giao vận và tầng ứng dụng luôn được bảo vệ an toàn bằng hàm băm, do vậy chúng không thể nào thay đổi được (chẳng hạn bằng cách phiên dịch số hiệu cổng). Chế độ vận chuyển sử dụng cho truyền thông từ máy khách đến máy khách.

Một phương tiện đóng gói các thông điệp IPsec dùng trong phần mềm biến đổi địa chỉ mạng NAT (*Network Address Translation*) đã được định nghĩa bởi các tài liệu RFC, được mô tả trong cơ chế NAT-T.

Chế độ đường ống

Trong chế độ đường ống, toàn bộ gói tin IP được mã hóa và/hoặc được nhận dạng. Khi đó ta đóng gói nó thành một gói tin IP mới với một tiêu đề IP mới. Chế độ đường ống được dùng để tạo ra một mạng riêng ảo VPN (*Virtual Private Network*) sử dụng cả trong truyền thông từ mạng máy tính đến mạng máy tính (nghĩa là giữa các bộ định tuyến để kết nối các miền thông tin), trong truyền thông máy khách đến mạng máy tính (nghĩa là sự truy cập của người sử dụng ở xa) cũng như trong truyền thông máy khách đến máy khách (nghĩa là hội thoại cá nhân: *chat*). Chế độ đường ống cũng hỗ trợ NAT.

6.6.6. Sự phát triển

IPsec được phát triển gắn với IPv6 do vậy mọi thực hiện của nó phải hoàn tương thích với thực hiện của IPv6 nhưng mặt khác Ipv6 lại là một sự mở rộng không ràng buộc của Ipv4. Tuy nhiên do việc triển khai Ipv6 quá chậm nên IPsec thông thường được sử dụng nhiều hơn để bảo vệ truyền thông trên IPv4. Các giao thức IPsec ban đầu được xác định trong RFC 1825 and RFC 1829, được công bố năm 1995.

Đến năm 1998, các tài liệu đó được thay thế bởi RFC 2401 và RFC 2412 có vẻ không tương thích với các phiên bản cũ tuy về quan điểm thì đồng nhất.

Tiếp đó một giao thức nhận dạng tương hỗ và trao đổi khóa IKE (*Internet Key Exchange*) lại được xác định nhằm tạo ra và quản lý các tổ hợp an ninh. Tháng Chạp năm 2005, những chuẩn mới được định nghĩa trong RFC 4301 and RFC 4309 đã thay thế rộng rãi các phiên bản cũ với một phiên bản mới của chuẩn trao đổi khóa là IKEv2.

Từ giữa năm 2008, một nhóm công tác bảo trì và phát triển đã thường xuyên hoạt động tại IETF.

Các thuật toán mã hóa

Các thuật toán mã hóa được xác định để dử dụng với IPsec gồm:

- HMAC-SHA1 để bảo vệ toàn vẹn thông tin và để nhận dạng
- TripleDES-CBC để bảo mật thông tin
- AES-CBC cũng để bảo mật thông tin.

Có thể tra cứu chi tiết trong RFC 4835.

Các công cụ phần mềm

Hỗ trợ IPsec thường được thực hiện trong "*hạt nhân*" với một phần mềm quản lý khóa và một quy trình thương lượng ISAKMP/IKE.

Có nhiều công cụ thực hiện các giao thức IPsec và ISAKMP/IKE như là:

- OpenBSD, với mã riêng của nó phát triển từ một công cụ BSD/OS được viết bởi John Ioannidis and Angelos D. Keromytis vào năm 1996.
- NRL IPsec, một trong những nguồn gốc của mã IPsec.
- Cụm phần mềm KAME, bao gồm Mac OS X, NetBSD and FreeBSD.
- "IPsec" trong Cisco IOS Software
- "IPsec" trong Microsoft Windows, bao gồm cả Windows XP, Windows 2000, Windows 2003, Windows Vista, Windows Server 2008, và Windows 7.
- IPsec trong Windows Vista và sau nữa.
- Bộ công cụ Authentec QuickSec
- IPsec trong Solaris
- Hệ điều hành IBM AIX
- IBM z/OS
- IPsec và IKE trong HP-UX (HP-UX IPsec)
- Cụm phần mềm Linux IPsec do Alexey Kuznetsov và David S. Miller viết.
- Openswan trên Linux, FreeBSD và Mac OS X sử dụng cụm phần mềm sinh Linux IPsec stack, hoặc cụm phần mềm KLIPS của chính nó.
- StrongSwan trên Linux, FreeBSD, Mac OS X, và Android sử dụng cụm phần mềm sinh IPsec.

PHẦN PHỤ LỤC

Phụ lục 1

1. HÀM LOGIC XOR

Hàm tuyển ngặt (*Exclusive OR*) hay là hàm cộng modulo 2 là một trong những hàm cơ bản được sử dụng khá phổ biến trong mật mã học (cũng như trong nhiều ứng dụng khác). Nhà toán học Anh George Boole ở cuối thế kỷ XIX đã sáng lập ra một ngành "Đại số học" mà sau này đã trở thành nền tảng cho việc xây dựng chế tạo các máy tính điện tử và các chip vi điện tử. Boole đã định nghĩa một số hàm logic hai biến cơ bản dạng: f = f(x,y) trong đó x, y (biến đầu vào: *input)* và f (biến đầu ra: *output*) đều là những biến logic, nghĩa là những biến số chỉ lấy giá trị trong tập hợp {0, 1} với 0 là giá trị phi lý - giá trị sai, còn 1 là giá trị chân lý - giá trị đúng.

Các hàm Boole một và hai biến cơ bản thông dụng nhất là:

NOT

Hàm phủ định của biến đầu vào a là ā. Giá trị của biến đầu ra ā đối lập với biến đầu vào a. *(Ý nghĩa:* ā luôn trái ngược với a)

a	**ā**
0	1
1	0

AND

Hàm hội f = a ∩ b. Giá trị của biến đầu ra f bằng 1 khi và chỉ khi cả hai biến đầu vào cùng có giá trị bằng 1, các trường hợp còn lại f lấy giá trị bằng 0. (*Ý nghĩa*: f đúng khi và chỉ khi vừa a vừa b cùng đúng)

a	**b**	**a ∩ b**
0	0	0
0	1	0
1	0	0
1	1	1

OR

Hàm tuyển f = a ∪ b. f bằng 1 khi và chỉ khi có hoặc a hoặc b, hoặc cả a và b có giá trị bằng 1. (*Ý nghĩa*: f đúng khi hoặc a đúng hoặc b đúng hoặc cả hai a và b cùng đúng)

a	**b**	**a ∪ b**
0	0	0
0	1	1
1	0	1
1	1	1

XOR

Hàm tuyển ngặt (*Exclusive OR*) hay còn gọi là hàm cộng modulo 2: f = p XOR q hay f = p ⊕ q. f lấy giá trị 1 khi và chỉ khi chỉ có một trong hai biến p hoặc q có giá trị bằng 1. (*Ý nghĩa:* nếu cả a và b cùng sai hay cùng đúng thì f sai)

a	**b**	**a XOR b**
0	0	0
0	1	1
1	0	1
1	1	0

Ba hàm logic NOT, AND và OR quá quen thuộc trong đại số logic và ý nghĩa rất rõ ràng. Ta chỉ nói thêm về hàm XOR. Theo định nghĩa thì hàm XOR chỉ có giá trị bằng 1 khi mà một trong hai biến đầu vào của nó bằng 1. Ý nghĩa của hàm XOR liên quan đến "tính chất đồng nhất" của các biến đầu vào: Nếu hai biến đầu vào có cùng giá trị thì XOR là sai, còn nếu hai biến đầu vào khác giá trị nhau thì XOR là đúng.

Ứng dụng trong các thuật toán lập mã: Giả sử ta lấy bản plaintext P và XOR nó với một chìa khóa K: nó sẽ biến thành một bản ciphertext C trong đó có một số bit đã thay đổi. Nếu ta lại lấy ciphertext XOR với chìa khóa K ấy một lần nữa thì ta được lại plaintext.

Ví dụ:

Plaintext:	P =	100110001
Khóa K:	K =	001111001
Mã hóa: Ciphertext	C = P XOR K =	101001000
Giải mã: Plaintext	P = C XOR K =	100110001

Nhận xét:

- Độ dài (kích thước số bit) của khóa K rõ ràng có tác động rất lớn đến khả năng bảo mật của mã đối xứng. Chẳng hạn trong mã block có kích thước block là 56 - 64 bit thì ta dùng khóa K có kích thước cũng là 56 - 64 bit. Vì mỗi vị trí trong khóa có thể tùy chọn 1 trong 2 giá trị 0 hay 1 nên có tất cả: 2^{56} cách tạo khóa khác nhau! Đây là một con số rất lớn cho nên thông thường nguy cơ bị tấn công bạo lực thấp.

- Tuy nhiên vì phép toán XOR được thực hiện hoàn toàn đơn giản nên tốc độ lập mã, giải mã vẫn khá nhanh.

2. TÍNH TOÁN THỰC HÀNH MÃ SỬA SAI HAMMING

Trong mã đối xứng DEA ta có nhắc đến khái niệm bit dư ngang bậc (*extra parity bits*) để sửa sai. Bạn đọc đã quen biết với việc mã hóa thông tin trong truyền thông điện tử, với các giao thức trong họ TCP/IP chắc chắn đều đã biết về những phương pháp mã hóa phát hiện sai (*Error detection*) và mã hóa tự sửa sai (*Error correction*) chẳng hạn như mã Hamming.

Dưới đây chỉ nhắc lại tóm tắt phương pháp tính toán thực hành cụ thể.

Nguyên tắc mấu chốt của mã Hamming là việc sử dụng một số "bit dư ngang bậc" (*extra parity bit*) để nhận diện ra một bit sai trong thông tin được truyền đi.

Xuất phát từ một từ "mang thông tin", một từ mã tương ứng được tạo như sau:

1. Đánh dấu mọi bit tại các vị trí là lũy thừa của 2 để làm *bit dư ngang bậc* (Các vị trí thứ 1, 2, 4, 8, 16, 32, 64…)
2. Tất cả các bit ở những vị trí khác còn lại dùng để ghi thông tin gốc cần truyền (Vị trí thứ 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17,…)
3. Mỗi bit dư ngang bậc được tính toán theo giá trị của một số bit thông tin trong thông điệp. Vị trí của bit dư ngang bậc xác định dãy các bit lần lượt được xét hay bỏ qua:

- *Vị trí 1*: Xét 1 bit, bỏ qua 1 bit, xét 1 bit, bỏ qua 1 bit,v.v. (1, 3, 5, 7, 9, 11, 13, 15…)
- *Vị trí 2*: Xét 2 bit, bỏ qua 2 bit, xét 2 bit, bỏ qua 2 bit,v.v. (2, 3, 6, 7, 10, 11, 14, 15…)
- *Vị trí 4*: Xét 4 bit, bỏ qua 4 bit,v.v. (4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23, …)

- *Vị trí 8*: Xét 8 bit, bỏ qua 8 bit, …(8 - 15, 24 - 31, 40 - 63, 80 - 95, …)
- *Vị trí 16*: Xét 16 bit, bỏ qua 16 bit, …(16 - 31, 48 - 63, 80 - 95, …)

4. Chọn lấy giá trị của bit dư parity là 1 nếu số các ký tự số 1 tại các vị trí được xét của nó là lẻ. Chọn lấy giá trị của bit dư parity là 0 nếu số các ký tự số 1 tại các vị trí được xét của nó là chẵn.

Ví dụ: Một thông điệp cần truyền là: 10011010

Tạo một thông điệp dữ liệu để truyền thông điệp đó, giành các vị trí thích hợp cho các bit dư:

_ _ 1 _ 0 0 1 _ 1 0 1 0

Tính giá trị cho từng bit dư parity (Dấu ? ký hiệu cho giá trị của bit parity cần tính)

- Vị trí 1 các bit được xét là 1, 3, 5, 7, 9, 11:
  ? _ 1 _ 0 0 1 _ 1 0 1 0. Có 4 số 1: chẵn, vậy giá trị bit parity ở vị trí 1 là 0: 0 _ 1 _ 0 0 1 _ 1 0 1 0
- Vị trí 2, các bit được xét là: 2, 3, 6, 7, 10, 11:
  0 ? 1 _ 0 0 1 _ 1 0 1 0. Có 3 số 1: lẻ, vậy giá trị bit parity ở vị trí 2 là 1: 0 1 1 _ 0 0 1 _ 1 0 1 0
- Vị trí 4 các bit được xét là: 4, 5, 6, 7, 12:
  0 1 1 ? 0 0 1 _ 1 0 1 0. Số lẻ: giá trị parity là 1:
  0 1 1 1 0 0 1 _ 1 0 1 0
- Vị trí 8, các bit được xét là: 8, 9, 10, 11, 12:
  0 1 1 1 0 0 1 ? 1 0 1 0. Số chẵn: giá trị parity là 0:
  0 1 1 1 0 0 1 0 1 0 1 0
- Thông điệp mã hóa: 011100101010.

Phát hiện bit sai

Việc sử dụng bit parity đơn cho phép bạn phát hiện là tồn tại một bit sai trong thông điệp nhận được. Muốn sửa sai ta cần biết thêm vị trí cụ thể của bit sai đó trong thông điệp vì nếu chỉ biết là trong thông điệp có một bit sai mà không biết vị trí của nó thì không thể sửa được.

Nếu có nhiều bit dư parity gắn vào một thông điệp và các bit dư đó được tổ chức sao cho sự hiện diện của các bit sai tại các vị trí khác nhau sẽ cho ra những kết quả tính toán sai khác nhau thì các bit sai có thể nhận diện được. Trong một thông điệp 7 bit chẳng hạn, vị trí của bit sai có thể có 7 khả năng, vì vậy với 3 bit kiểm tra là ta đã có thể phát hiện không những là có 1 bit sai mà còn có thể định vị bit sai đó. Tương tự như vậy, nếu một họ từ mã hóa được chọn sao cho khoảng cách bé nhất giữa các từ đó ít nhất bằng 3 thì mã sửa sai với 1 bit là có thể. Việc tiếp cận theo khoảng cách đó có tính chất "hình học", trong khi lập luận tính toán bit sai trên kia có tính chất "đại số". Những lập luận trên đây dùng để dẫn dắt ta đến khái niệm về Mã tự sửa sai Hamming, một phương pháp kiểm tra cho phép bạn tự sửa 1 sai.

Chẳng hạn, trong ví dụ trên cho ta thông điệp mã hóa là: 011100101010. Bên A gửi thông điệp đó đi. Giả sử do một lý do nào đó trong quá trình truyền tin, thông điệp bên B nhận được lại là: 011100101110. Người nhận (B) sẽ căn cứ vào việc kiểm tra các bit dư parity để phát hiện xem trong thông điệp có bit nào sai và do đó có thể tự sửa sai. B sẽ tính lại từng bit kiểm tra trong thông điệp nhận được bằng phương pháp như trước. Làm như vậy ta thấy ngay các bit parity thứ 2 và thứ 8 là sai! Vậy thì: 2 + 8 = 10. 10 là vị trí của bit thông tin bị sai (số 1 trong thông điệp nhận được ở vị trí thứ 10 là sai, cần sửa thành số 0). *Trong trường hợp tổng quát, kiểm tra lại tất cả các bit parity sai, tổng của các vị trí của chúng cho ta vị trí của bit thông tin bị sai.*

Bạn hãy tự kiểm tra bằng phương pháp mã Hamming xem trong các thông điệp nhận được sau đây, thông điệp nào có sai ở vị trí nào và cần được sửa lại như thế nào?

- 010101100011
- 111110001100
- 000010001010

Phụ lục 2

1. HÀM MODULO - ĐỒNG DƯ THỨC

Hàm modulo có thể hiểu một cách đơn giản chính là số dư trong phép chia các số nguyên. Muốn tính X modulo Y (thường ký hiệu là *X mod Y*) ta chỉ cần làm phép chia X cho Y và tìm số dư trong phép chia đó, nói khác đi: ta trừ vào X bội số lớn nhất của Y bé hơn X. Rõ ràng X mod Y chỉ có thể lấy các giá trị từ 0, 1,… cho đến Y-1.

Ví dụ:

25 mod 5 = 0

15 mod 7 = 1

33 mod 12 = 9

203 mod 256 = 203

Trong số học, hai số nguyên A và B được gọi là "đồng dư theo modulo N" nếu chúng có cùng số dư trong phép chia cho N. Ta ký hiệu: A ≡ B (mod N) và đọc là "*A đồng dư với B theo modulo N*". Biểu thức đó gọi là một đồng dư thức.

Ví dụ: 18 ≡ 4 (mod 7) ≡ 11 (mod 7)

Hàm modulo trong số học rất hữu ích trong các thuật toán mật mã vì nó cho phép chúng ta xác định kích thước của một phép toán và do đó chắc chắn là không có kết quả là những con số quá lớn. Đây là một nhận xét rất quan trọng khi sử dụng máy tính kỹ thuật số. Hàm modulo được dùng trong thuật toán RSA lập khóa mã công khai và khóa mã riêng.

2. GIẢI THUẬT EUCLID

Giải thuật Euclid, hay thuật toán Euclid, là một giải thuật tính ước số chung lớn nhất (USCLN) của hai số (nguyên) một cách hiệu quả. Giải thuật này được biết đến từ khoảng năm 300 trước Công Nguyên. Nhà toán học Cổ Hy Lạp Euclid đã nêu giải thuật này trong cuốn sách "Cơ sở" (*Elements*) nổi tiếng.

Ví dụ: Tính ước số chung lớn nhất của 91 và 287.

Trước hết lấy 287 (số lớn hơn trong 2 số) chia cho 91:

287 = 91*3+14 (91 và 14 sẽ được dùng cho vòng lặp kế tiếp)

Nhận xét: Bất kỳ số nào chia hết bởi 287 và 91 cũng sẽ chia hết bởi 287 - 91*3 = 14. Tương tự, số chia hết bởi 91 và 14 cũng chia hết bởi 91*3 + 14 = 287. Do đó, USCLN(91,287) = USCLN(91,14). Bài toán trở thành tìm USCLN(91,14). Lặp lại quy trình trên cho đến khi phép chia không còn số dư nữa. 91 = 14*6 + 7 (14 và 7 sẽ được dùng cho vòng lặp kế tiếp) 14 = 7*2 + 0 (không còn số dư, kết thúc, nhận 7 làm kết quả).

Cuối cùng ta có:

7 = USCLN(14,7) = USCLN(91,14) = USCLN(287,91).

Bổ đề. Giả sử a = bq + r, với a, b, q, r là các số nguyên, ta có:

$$\text{UCLN}(a,b) = \begin{cases} b & \text{nếu } r = 0 \\ \text{UCLN}(b,r) & \text{nếu } r \neq 0 \end{cases}$$

Mã giải:

Chương trình đệ quy procedure USCLN(a, b: positive integers)

```
Begin
   if a mod b = 0 then USCLN:= b
   else USCLN(b; a mod b);
End
```

Chương trình dùng vòng lặp procedure USCLN(a, b: positive integers)

```
Begin
    x:= a
    y:= b
while y ≠ 0
begin
    r:= x mod y
    x:= y
    y:= r
End {x là USCLN cần tìm}
End
```

3. GIẢI THUẬT EUCLID MỞ RỘNG

Giải thuật Euclid mở rộng sử dụng để giải phương trình vô định nguyên (còn được gọi là phương trình Đi-ô-phăng)

a*x + b*y = c

trong đó *a, b, c* là các hệ số nguyên, *x, y* là các ẩn nhận giá trị nguyên. Điều kiện cần và đủ để phương trình này có nghiệm (nguyên) là *UCLN(a, b)* là ước của *c*.

Khẳng định này dựa trên mệnh đề sau trong số học:

> "Ta biết rằng nếu d = USCLN(a,b) thì tồn tại các số nguyên x, y sao cho: a*x + b*y = d."

Cơ sở lý thuyết của giải thuật

Giải thuật Euclid mở rộng kết hợp quá trình tìm UCLN*(a, b)* trong thuật toán Euclid với việc tìm một cặp số x, y thỏa mãn phương trình Đi-ô-phăng. Giả sử cho hai số tự nhiên a, b, ngoài ra a > b > 0. Đặt $r_o = a$, $r_1 = b$, chia r_0 cho r_1 được số dư r_2. Nếu $r_2 = 0$ thì dừng, nếu r_2 khác không, chia r_1 cho r_2 được số dư r_3,... Vì dãy các r_i là giảm thực sự nên sau hữu hạn bước ta được số dư $r_m = 0$.

$r_o = q_1 * r_1 + r_2, 0 < r_2 < r_1;$

$r_1 = q_2 * r_2 + r_3, 0 < r_3 < r_2;$

....

$r_{m-1} = q_m * r_m + r_{m+1}, 0 < r_{m+1} < r_m;$

$r_m = q_{m+1} * r_{m+1};$

trong đó số dư cuối cùng khác 0 là $r_{m+1} = d$.

Bài toán đặt ra là tìm x, y sao cho: $a * x + b * y = r_{m+1}(= d)$

Để làm điều này, ta tìm x, y theo công thức truy hồi, nghĩa là tìm x_i và y_i sao cho: $a * x_i + b * y_i = r_i$ với $i = 0, 1,....$

Ta có:

$a * 1 + b * 0 = a = r_o$ và $a * 0 + b * 1 = b = r_1$,

nghĩa là:

$x_o = 1, x_1 = 0$ và $y_o = 0, y_1 = 1$ (1)

Tổng quát, giả sử có:

$a * x_i + b * y_i = r_i$

với i = 0,1,.... $a * x_{i+1} + b * y_{i+1} = r_{i+1}$ với i = 0,1,....

Khi đó từ: $r_i = q_{i+1} * r_{i+1} + r_{i+2}$

suy ra:

$r_i - q_{i+1} * r_{i+1} = r_{i+2}$

$(a * x_i + b * y_i) - q_{i+1} * (a * x_{i+1} + b * y_{i+1}) = r_{i+2}$

$a * (x_i - q_{i+1} * x_{i+1}) + b * (y_i - q_{i+1} * y_{i+1}) = r_{i+2}$

từ đó, có thể chọn:

$x_{i+2} = x_i - q_{i+1} * x_{i+1}$ (2)

$y_{i+2} = y_i - q_{i+1} * y_{i+1}$ (3)

Khi i = m - 1 ta có được x_{m+1} và y_{m+1}.

Các công thức (1), (2), (3) là công thức truy hồi để tính x, y.

Giải thuật

Giải thuật sau chỉ thực hiện với các số nguyên a > b > 0, biểu diễn bằng:

```
Procedure Euclid_Extended (a,b)
Var Int x0:=1, x1:=0, y0=0,y1:=1;
While b>0
do      {r:= a mod b
         q:= a div b
         x:= x0-x1*q
         y:= y0-y1*q
         if r=0 then Break
         a:=b
         b:=r
         x0:=x1
         x1:=x
         y0:=y1
         y1:=y}
Return d:=b, x, y;
```

Ví dụ:

Giả sử cho a = 29, b = 8, giải thuật trải qua các bước như sau:

Bước i	r_i	r_{i+1}	r_{i+2}	q_{i+1}	x_i	x_{i+1}	x_{i+2}	y_i	y_{i+1}	y_{i+2}
0	29	8	5	3	1	0	1	0	1	-3
1	8	5	3	1	0	1	-1	1	-3	4
2	5	3	2	1	1	-1	2	-3	4	-7
3	3	2	1	1	-1	2	-3	4	-7	11
4	2	1	0	2						

Kết quả thuật toán cho đồng thời:

$d = UCLN(29,8) = 1$ và $x = -3$, $y = 11$.

Dễ dàng kiểm tra hệ thức $29 * (-3) + 8 * 11 = 1$

Áp dụng giải thuật Euclid mở rộng tìm số nghịch đảo trong vành $\mathbb{Z}_m$

Số nghịch đảo trong vành $\mathbb{Z}_m$

Trong lý thuyết số, vành $\mathbb{Z}_m$ được định nghĩa là vành thương của $\mathbb{Z}$ (vành các số nguyên) với quan hệ đồng dư theo modulo m (là một quan hệ tương đương) mà các phần tử của nó là các lớp đồng dư theo modulo m (m là một số nguyên dương lớn hơn 1). Ta cũng có thể xét $\mathbb{Z}_m$ chỉ với các đại diện của nó.

Khi đó:

$$\mathbb{Z}_m = \{0, 1, ..., m-1\}$$

Phép cộng và nhân trong $\mathbb{Z}_m$ là phép toán thông thường rút gọn theo modulo m:

$$a + b = (a + b) \bmod m$$
$$a * b = (a * b) \bmod m$$

Phần tử a của $\mathbb{Z}_m$ được gọi là khả đảo trong $\mathbb{Z}_m$ hay khả đảo theo modulo m nếu tồn tại phần tử a' trong $\mathbb{Z}_m$ sao cho a*a' = 1 trong $\mathbb{Z}_m$. Khi đó a' được gọi là nghịch đảo modulo m của a. Trong lý thuyết số đã chứng minh rằng, số a là khả đảo theo modulo m khi và chỉ khi USCLN của a và m bằng 1 (a và m nguyên tố cùng nhau). Khi đó tồn tại các số nguyên x, y sao cho: m * x + a * y = 1.

Đẳng thức này lại chỉ ra y là nghịch đảo của a theo modulo m. Do đó có thể tìm được phần tử nghịch đảo của a theo modulo m nhờ thuật toán Euclid mở rộng khi chia m cho a.

Giải thuật

Giải thuật sau chỉ thực hiện với các số nguyên m > a > 0, biểu diễn bằng dãy mã:

```
Procedure Euclid_Extended (a,m)
int, y0=0,y1:=1;
While a>0
do {r:= m mod a
 if r=0 then Break
q:= m div a
y:= y0-y1*q
m:=a
a:=r
y0:=y1
y1:=y}
If a>1 Then Return "A không khả nghịch theo modulo m"
else Return " Nghịch đảo modulo m của a là y"
```

Ví dụ: Tìm số nghịch đảo (nếu có) của 30 theo mô-đun 101

Bước i	**m**	**a**	**r**	**q**	**y0**	**y1**	**y**
0	101	30	11	3	0	1	-3
1	30	11	8	2	1	-3	7
2	11	8	3	1	-3	7	-10
3	8	3	2	2	7	-10	27
4	3	2	1	1	-10	27	-37
5	2	1	0	.	.	.	.

Kết quả tính toán trong bảng cho ta -37. Lấy số đối của 37 theo mô-đun 101 được 64. Vậy $30^{-1} \bmod 101 = 64$.

Ứng dụng

Số nghịch đảo theo modulo được ứng dụng nhiều trong việc giải phương trình đồng dư, trong lý thuyết mật mã, đặc biệt trong thuật toán RSA.

4. ĐỊNH LÝ SỐ DƯ TRUNG QUỐC

Định lý số dư Trung Quốc (Chinese Theorem of Remainders) là tên người phương tây đặt cho định lý này. Người Trung Quốc gọi nó là bài toán Hàn Tín điểm binh.

Hàn Tín là một danh tướng thời Hán Sở từng được phong tước vương thời Hán Cao Tổ Lưu Bang dựng nghiệp. Sử ký của Tư Mã Thiên viết rằng Hàn Tín là tướng trói gà không nổi, nhưng rất có tài quân sự. Tương truyền rằng khi Hàn Tín điểm quân, ông cho quân lính xếp hàng 3, hàng 5, hàng 7 rồi báo cáo số dư. Từ đó ông tính chính xác quân số đến từng người.

Gần đây, định lý số dư Trung Quốc có nhiều ứng dụng trong các bài toán về số nguyên lớn áp dụng vào lý thuyết mật mã.

Định lý: Cho n số nguyên dương m_1, m_2, m_3,…, m_n đôi một nguyên tố cùng nhau. Khi đó hệ đồng dư tuyến tính:

$$\begin{cases} x \equiv a_i (\mathrm{mod}\, m_i) \\ i = \overline{1,n} \end{cases}$$

có nghiệm duy nhất mô-đun $M = m_1 m_2 \ldots m_m$.

Định lý số dư Trung Quốc khẳng định về sự tồn tại duy nhất của một lớp thặng dư các số nguyên thỏa mãn đồng thời nhiều đồng dư thức tuyến tính. Do đó có thể sử dụng định lý để giải quyết những bài toán về sự tồn tại và đếm các số nguyên thỏa mãn một hệ các điều

kiện quan hệ đồng dư, chia hết…, hay đếm số nghiệm của phương trình đồng dư. Bản chất của bài toán Hàn Tín điểm binh là việc giải hệ phương trình đồng dư bậc nhất.

$$\begin{cases} x \equiv a_1 (\mathrm{mod}\, m_1) \\ x \equiv a_2 (\mathrm{mod}\, m_2) \\ \ldots \\ x \equiv a_k (\mathrm{mod}\, m_k) \end{cases}$$

trong đó $m_1, m_2, \ldots, m_k$ đôi một nguyên tố cùng nhau.

Hệ phương trình đồng dư nói trên có nghiệm duy nhất theo mô-đun $M = m_1.m_2 \ldots m_k$ là:

$$x \equiv a_1.M_1.y_1 + a_2.M_2.y_2 + \ldots + a_k.M_k.y_k \pmod{M}$$

trong đó:

$$M_1 = M/m_1,\ M_2 = M/m_2, \ldots,\ M_k = M/m_k$$

và:

$$y_1 = (M_1)^{-1} (\mathrm{mod}\, m_1),$$

$$y_2 = (M_2)^{-1} (\mathrm{mod}\, m_2),$$

...

$$y_k = (M_k)^{-1} (\mathrm{mod}\, m_k)$$

trong đó:

$(M_1)^{-1} (\mathrm{mod}\, m_1)$ là nghịch đảo theo modulo của m_1

với: $y_1 = (M_1)^{-1} (\mathrm{mod}\, m_1) \Leftrightarrow y_1 M_1 = 1 (\mathrm{mod}\, m_1)$

Ví dụ: Một đội quân, nếu xếp hàng 3 thì dư ra 2 người, xếp hàng 5 thì dư ra 3 người còn xếp hàng 7 thì dư ra 5 người. Hãy tính chính xác quân số *x* của đội quân đó.

Giải hệ phương trình đồng dư:

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 5 \pmod 7 \end{cases}$$

ta có:

$M = 3.5.7 = 105$; $M_1 = 5.7 = 35$,

$M_2 = 3.7 = 21$, $M_3 = 3.5 = 15$.

$y_1 = 35 - 1 \pmod 3 = 2 - 1 \pmod 3 = 2$;

$y_2 = 21 - 1 \pmod 5 = 1 - 1 \pmod 5 = 1$;

$y_3 = 15 - 1 \pmod 7 = 1 - 1 \pmod 7 = 1$.

Từ đó:

$$x \equiv 2.35.2 + 3.21.1 + 5.15.1 \pmod{105}$$
$$x \equiv 140 + 63 + 75 \pmod{105} \equiv 278 \pmod{105}$$
$$x \equiv 68 \pmod{105}$$

Như vậy x có dạng $x = 68 + k.105$, k là số nguyên bất kỳ (hoặc số nguyên thích hợp nếu tìm nghiệm thỏa mãn một số ràng buộc phụ nào đấy).

Còn lưu truyền lại một phát biểu của công thức giải bài toán Hàn Tín điểm binh có dạng một *"khẩu quyết"* khó hiểu được lưu truyền cho đến nay dưới dạng một bài thơ thất ngôn tứ tuyệt là:

* *Tam (3)* nhân đồng hành*, thất thập (70)* hi ⇔ 3 người cùng đi, hiếm kẻ 70 tuổi, hiểu là: lấy số dư khi xếp hàng 3 nhân cho 70

* *Ngũ (5)* thụ mai hoa, *trấp nhất (21)* chi ⇔ 5 cây hoa mai có 21 cành, hiểu là: lấy số dư khi xếp hàng 5 nhân cho 21

* *Thất (7)* tử đoàn viên chính *bán nguyệt (15)* ⇔ 7 đứa con sum vầy trong ngày 15, hiểu là: lấy số dư khi xếp hàng 7 nhân cho 15

* *Gia bách linh ngũ (105)* định vi kỳ ⇔ thêm vào 105 thì được số phải tìm.

Thực ra nghiệm của bài toán này là không duy nhất, phải có thêm ràng buộc ngoài, chẳng hạn ước tính số quân trong đơn vị là trong một khoảng nào đó, chẳng hạn trong ví dụ cụ thể sau đây:

Một đơn vị khoảng 200 - 300 quân, sau một trận đánh quay về, cần điểm lại xem quân số chính xác còn lại là bao nhiêu bằng cách xếp hàng 3, đếm số dư, xếp hàng 5, đếm số dư, xếp hàng 7 đếm số dư; ở đây ta lấy ***k = 2****: tính được quân số còn lại sau trận đánh là* ***278*** *(vì từ k ≥ 3 hoặc k ≤ 1 thì sai với điều kiện ràng buộc về quân số ban đầu là 200 – 300, chính ràng buộc này cho phép ta xác định nghiệm số duy nhất của bài toán).*

5. BÀI TOÁN XẾP BA LÔ

Bài toán xếp ba lô (một số sách ghi là bài toán cái túi) là một bài toán tối ưu hóa tổ hợp. Bài toán được đặt tên từ vấn đề chọn những gì quan trọng có thể nhét vừa vào trong một cái túi (với giới hạn khối lượng) để mang theo trong một chuyến đi. Các bài toán tương tự thường xuất hiện trong nhiều vấn đề của toán ứng dụng như: Bài toán lựa chọn phương án kinh doanh, các bài toán tổ hợp, lý thuyết độ phức tạp tính toán, mật mã học.

Phát biểu của bài toán thực tế

Một người đi xa chỉ có một cái túi (ba lô) có sức chứa tối đa về trọng lượng là *C*. Người đó có *n* mặt hàng, mỗi loại có trọng lượng và giá trị khác nhau, vậy người đó nên bỏ vào ba lô những loại hàng nào và mỗi loại với số lượng bao nhiêu để đạt tổng giá trị cao nhất trong khả năng có thể mang đi được.

Trong các phát biểu sau đây ta gọi x_j là số lượng đồ vật loại j, p_j là đơn giá của đồ vật loại j còn ωj là giá trị của một đơn vị loại j.

Bài xếp ba lô dạng 0-1

Hạn chế về số đồ vật thuộc mỗi loại là 0 (không được chọn) và 1 (được chọn). Bài xếp ba lô 0-1 có thể được phát biểu toán học như sau:

Cực đại hóa dạng tuyến tính:

$$\sum_{j=1}^{n} p_j x_j$$

với các điều kiện ràng buộc:

$$\sum_{j=1}^{n} w_j x_j \le c, \quad x_j = 0 \text{ hoặc } 1,\ j = 1,...,n$$

Bài xếp ba lô bị chặn

Hạn chế số đồ vật thuộc mỗi loại không được vượt quá một lượng nào đó. Bài xếp ba lô bị chặn có thể được phát biểu bằng toán học như sau:

Cực đại hóa:

$$\sum_{j=1}^{n} p_j x_j$$

Với các ràng buộc:

$$\sum_{j=1}^{n} w_j x_j \le c, \quad 0 \le x_j \le b_j,\ j = 1,...,n$$

Bài xếp ba lô không bị chặn

Không có một hạn chế nào về số đồ vật mỗi loại.

Một trường hợp đặc biệt của bài toán này nhận được nhiều quan tâm, đó là bài toán với các tính chất:

- Là một bài toán quyết định

- Là một bài toán 0/1

- Với mỗi đồ vật, chi phí bằng giá trị: $C = V$

Lưu ý rằng trong trường hợp đặc biệt này, bài toán tương đương với:

- Cho một tập các số nguyên, tồn tại hay không một tập con có tổng đúng bằng *C*?

- Hoặc nếu đồ vật được phép có chi phí âm và *C* được chọn bằng 0, bài toán có dạng: Cho trước một tập số nguyên, tồn tại hay không một tập con có tổng đúng bằng *0*?

Trường hợp đặc biệt này được gọi là bài toán tổng các tập con (*subset sum problem*). Với một số lý do, trong ngành mật mã học, người ta thường dùng cụm từ "bài toán xếp ba lô" khi thực ra đang có ý nói về "bài toán tổng con".

Bài toán xếp ba lô thường được giải bằng quy hoạch động, tuy chưa có một thuật toán thời gian đa thức cho bài toán tổng quát. Cả bài xếp ba lô tổng quát và bài toán tổng con đều là các bài NP-khó, và điều này dẫn đến các cố gắng sử dụng tổng con làm cơ sở cho các hệ thống mật mã hóa khóa công khai, chẳng hạn Merkle-Hellman. Các cố gắng này thường dùng nhóm thay vì các số nguyên. Merkle-Hellman và một số thuật toán tương tự khác đã bị phá, do các bài toán tổng con cụ thể mà họ tạo ra thực ra lại giải được bằng các thuật toán thời gian đa thức.

Phiên bản bài toán quyết định của bài xếp ba lô được mô tả ở trên là NP-đầy đủ và trong thực tế là một trong 21 bài toán NP-đầy đủ của Karp.

Bài xếp ba lô dạng phân số

Với mỗi loại, có thể chọn một phần của nó (ví dụ: 1kg bánh mì có thể được cắt ra thành nhiều phần để bỏ vào ba lô)

Cách giải bằng quy hoạch động

Bài toán xếp ba lô có thể được giải trong thời gian giả-đa thức bằng quy hoạch động. Dưới đây là lời giải quy hoạch động cho *bài toán xếp ba lô không bị chặn*.

Gọi các chi phí là $c_1,..., c_n$ và các giá trị tương ứng là $v_1,..., v_n$. Ta cần cực đại hóa tổng giá trị với điều kiện tổng chi phí không vượt quá C. Khi đó, với mỗi $i \leq C$, đặt $A(i)$ là giá trị lớn nhất có thể đạt được với tổng chi phí không vượt quá i. Rõ ràng, $A(C)$ là đáp số của bài toán.

Định nghĩa $A(i)$ một cách đệ quy như sau:

- $A(0) = 0$
- $A(i) = \max \{ v_j + A(i - c_j) \mid c_j \leq i \}$

Ở đây, giá trị lớn nhất của tập rỗng được lấy bằng 0. Tính dần các kết quả từ $A(0)$ tới $A(C)$, ta sẽ được lời giải. Do việc tính mỗi $A(i)$ đòi hỏi xem xét n đồ vật (tất cả các giá trị này đã được tính từ trước), và có C giá trị của các $A(i)$ cần tính, nên thời gian chạy của lời giải quy hoạch động là $O(nC)$. Điều này không mâu thuẫn với thực tế rằng bài toán xếp ba lô là NP-đầy đủ, do C, không như n, không thuộc mức đa thức theo độ dài của đầu vào cho bài toán. Độ dài đầu vào bài toán tỉ lệ thuận với số bit trong C, chứ không tỉ lệ với chính C.

Một giải pháp quy hoạch động tương tự cho *bài toán xếp ba lô 0-1* cũng chạy trong thời gian giả-đa thức. Cũng như trên, gọi các chi phí là $c_1,..., c_n$ và các giá trị tương ứng là $v_1,..., v_n$. Ta cần cực đại hóa tổng giá trị với điều kiện tổng chi phí không vượt quá C. Định nghĩa một hàm đệ quy $A(i, j)$ là giá trị lớn nhất có thể đạt được với chi phí không vượt quá j và sử dụng các đồ vật trong khoảng từ x_1 tới x_i.

$A(i,j)$ được định nghĩa đệ quy như sau:

- $A(0, j) = 0$
- $A(i, 0) = 0$
- $A(i, j) = A(i - 1, j)$ nếu $c_i > j$
- $A(i, j) = \max(A(i - 1, j), v_i + A(i - 1, j - c_i))$ nếu $c_i \leq j$

Để có lời giải, ta tính A(n, C). Để làm điều này, ta có thể dùng 1 bảng để lưu các tính toán trước đó. Cách giải này sẽ chạy trong thời gian O(nC) và không gian O(nC), tuy ta có thể giảm độ phức tạp không gian xuống O(C) bằng một số sửa đổi nhỏ.

Thuật toán tham lam

Martello và Toth (1990) đã đưa ra một thuật toán gần đúng kiểu tham lam (*greedy approximation algorithm*) để giải bài toán xếp ba lô. Giải thuật này sắp xếp các đồ vật theo thứ tự giảm dần về giá trị, sau đó theo thứ tự đó xếp các đồ vật vào ba lô cho đến khi không cho thêm được đồ vật nào vào nữa.

Phụ lục 3

THÔNG TƯ SỐ 09/2011/TT-BCT NGÀY 30/3/2011 CỦA BỘ CÔNG THƯƠNG

Quy định về việc quản lý, sử dụng chữ ký số, chứng thư số và dịch vụ chứng thực chữ ký số của bộ công thương

BỘ TRƯỞNG BỘ CÔNG THƯƠNG

- Căn cứ Nghị định số 189/2007/NĐ-CP ngày 27 tháng 12 năm 2007 của Chính phủ quy định chức năng, nhiệm vụ, quyền hạn và cơ cấu tổ chức của Bộ Công thương; Căn cứ Nghị quyết 59/NQ-CP về việc đơn giản hóa thủ tục hành chính thuộc phạm vi chức năng quản lý của Bộ Công thương;

- Căn cứ Nghị định số 26/2007/NĐ-CP ngày 15 tháng 02 năm 2007 của Chính phủ quy định chi tiết thi hành Luật Giao dịch điện tử về chữ ký số và dịch vụ chứng thực chữ ký số; Căn cứ Nghị định số 64/2007/NĐ-CP ngày 10 tháng 4 năm 2007 của Chính phủ quy định về ứng dụng công nghệ thông tin trong hoạt động của cơ quan nhà nước;

- Bộ trưởng Bộ Công thương quy định về việc quản lý, sử dụng chữ ký số, chứng thư số và dịch vụ chứng thực chữ ký số của Bộ Công thương như sau:

Chương I. QUY ĐỊNH CHUNG

Điều 1. Phạm vi điều chỉnh

Thông tư này quy định việc quản lý, sử dụng chữ ký số, chứng thư số và dịch vụ chứng thực chữ ký số trong giao dịch điện tử của Bộ Công thương.

Điều 2. Đối tượng áp dụng

1. Tổ chức, cá nhân thuộc Bộ Công thương, Sở Công thương các tỉnh, thành phố trực thuộc Trung ương.

2. Tổ chức, cá nhân khác lựa chọn sử dụng dịch vụ chữ ký số của Bộ Công thương trong các hoạt động giao dịch điện tử do Bộ Công thương tổ chức.

Điều 3. Giải thích từ ngữ: Trong Thông tư này, các từ ngữ dưới đây được hiểu như sau:

1. "Chứng thư số" là một dạng chứng thư điện tử do Tổ chức cung cấp dịch vụ chữ ký số của Bộ Công thương cấp.

2. "Chữ ký số" là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã không đối xứng theo đó người có được thông điệp dữ liệu ban đầu và khóa công khai của người ký có thể xác định được chính xác:

a) Việc biến đổi nêu trên được tạo ra bằng đúng khóa bí mật tương ứng với khóa công khai trong cùng một cặp khóa;

b) Sự toàn vẹn nội dung của thông điệp dữ liệu kể từ khi thực hiện việc biến đổi nêu trên.

3. "Dịch vụ chứng thực chữ ký số" là một loại hình dịch vụ do Tổ chức cung cấp dịch vụ chữ ký số của Bộ Công thương quản lý. Dịch vụ chứng thực chữ ký số bao gồm:

a) Tạo cặp khóa bao gồm khóa công khai và khóa bí mật cho thuê bao;

b) Cấp, gia hạn, tạm dừng, phục hồi và thu hồi chứng thư số của thuê bao;

c) Duy trì trực tuyến cơ sở dữ liệu về chứng thư số;

d) Những dịch vụ khác có liên quan theo quy định của Nghị định số 26/2007/NĐ-CP ngày 15 tháng 02 năm 2007 của Chính phủ quy định chi tiết thi hành Luật Giao dịch điện tử về chữ ký số và dịch vụ chứng thực chữ ký số (gọi tắt là Nghị định chữ ký số).

4. "Ký số" là việc đưa khóa bí mật vào một chương trình phần mềm để tự động tạo và gắn chữ ký số vào thông điệp dữ liệu.

5. "Người ký" là thuê bao dùng đúng khóa bí mật của mình để ký số vào một thông điệp dữ liệu.

6. "Người nhận" là tổ chức, cá nhân nhận được thông điệp dữ liệu được ký số bởi người ký, sử dụng chứng thư số của người ký đó để kiểm tra chữ ký số trong thông điệp dữ liệu nhận được và tiến hành các hoạt động, giao dịch có liên quan.

7. "Thuê bao" là tổ chức, cá nhân quy định tại Điều 2 Thông tư này; được Tổ chức cung cấp dịch vụ chữ ký số của Bộ Công thương cấp chứng thư số; chấp nhận chứng thư số và giữ khóa bí mật tương ứng với khóa công khai ghi trên chứng thư số được cấp.

8. "Tổ chức quản lý thuê bao" là các đơn vị thuộc Bộ Công thương, hoặc các tổ chức khác đề nghị cấp chứng thư số cho tổ chức, cá nhân thuộc tổ chức mình và chịu trách nhiệm theo quy định của pháp luật về quản lý tổ chức, cá nhân đó.

9. "Giao dịch điện tử của Bộ Công thương" là các hoạt động, nghiệp vụ được tiến hành bằng phương thức điện tử của Bộ Công thương.

Điều 4. Tổ chức cung cấp dịch vụ chữ ký số của Bộ Công thương

Tổ chức cung cấp dịch vụ chữ ký số của Bộ Công thương, do Cục Thương mại điện tử và Công nghệ thông tin quản lý, điều hành và là tổ chức duy nhất của Bộ Công thương cung cấp dịch vụ chứng thực chữ ký số.

Điều 5. Chứng thư số

1. Nội dung chứng thư số: Chứng thư số do Tổ chức cung cấp dịch vụ chữ ký số của Bộ Công thương quản lý phải bao gồm các nội dung sau:

a) Tên tổ chức cung cấp dịch vụ chữ ký số;

b) Tên thuê bao;

c) Tên tổ chức quản lý thuê bao;

d) Số hiệu của chứng thư số;

đ) Thời hạn có hiệu lực của chứng thư số;

e) Khóa công khai của thuê bao;

g) Chữ ký số của tổ chức cung cấp dịch vụ chữ ký số;

h) Các hạn chế về mục đích, phạm vi sử dụng của chứng thư số;

i) Các hạn chế về trách nhiệm pháp lý của Tổ chức cung cấp dịch vụ chữ ký số;

k) Các thông tin khác cho mục đích quản lý, sử dụng, an toàn, bảo mật do Tổ chức cung cấp dịch vụ chữ ký số quy định.

2. Thời gian có hiệu lực của chứng thư số: Không quá 05 (năm) năm đối với chứng thư số của thuê bao.

Chương II. CHỨC NĂNG, NHIỆM VỤ CỦA TỔ CHỨC CUNG CẤP DỊCH VỤ CHỮ KÝ SỐ, QUYỀN VÀ NGHĨA VỤ CỦA CÁC ĐỐI TƯỢNG SỬ DỤNG DỊCH VỤ CHỮ KÝ SỐ

Điều 6. Chức năng, nhiệm vụ của Tổ chức cung cấp dịch vụ chữ ký số

1. Quản lý việc cấp, gia hạn, tạm dừng, thu hồi, khôi phục chứng thư số và thay đổi cặp khóa cho thuê bao khi có yêu cầu. Hình thành và phát triển dịch vụ bảo đảm an toàn và an ninh thông tin; cung cấp dịch vụ chữ ký số.

2. Quản lý, vận hành hệ thống trang thiết bị kỹ thuật cung cấp dịch vụ chứng thực chữ ký số của Bộ Công thương, nghiên cứu, nâng cấp, đảm bảo duy trì hoạt động cung cấp dịch vụ chứng thực chữ ký số của Bộ Công thương an toàn, liên tục. Thử nghiệm và đề xuất ứng dụng các công nghệ mới để đảm bảo an ninh, an toàn thông tin phục vụ giao dịch điện tử.

3. Lưu trữ đầy đủ, chính xác và cập nhật thông tin của thuê bao phục vụ việc quản lý chứng thư số trong suốt thời gian chứng thư số

có hiệu lực. Trong trường hợp chứng thư bị thu hồi thì phải lưu trữ các thông tin chứng thư số của thuê bao trong thời hạn ít nhất 05 năm kể từ khi chứng thư số bị thu hồi.

4. Tổ chức cung cấp dịch vụ chữ ký số có chức năng chứng thực các chữ ký số lưu hành trên các văn bản, tài liệu điện tử và trong các giao dịch điện tử.

5. Hướng dẫn các tổ chức quản lý thuê bao, thuê bao thực hiện đúng các quy định tại Thông tư này.

Điều 7. Quyền và nghĩa vụ của tổ chức quản lý thuê bao

1. Được cung cấp thông tin hướng dẫn về trình tự, thủ tục cấp phát, quản lý và sử dụng chứng thư số.

2. Được yêu cầu Tổ chức cung cấp dịch vụ chữ ký số cấp, gia hạn, tạm dừng, khôi phục, thu hồi chứng thư số hoặc thay đổi cặp khóa cho các thuê bao do mình quản lý.

3. Chịu trách nhiệm về tính chính xác của các thông tin trên giấy đề nghị cấp, gia hạn, tạm dừng, khôi phục, thu hồi chứng thư số và thay đổi cặp khóa của thuê bao do mình quản lý.

4. Hướng dẫn, kiểm tra các thuê bao thuộc tổ chức mình quản lý, sử dụng chứng thư số và khóa bí mật theo đúng các quy định tại Thông tư này.

5. Thông báo kịp thời bằng văn bản cho Tổ chức cung cấp dịch vụ chữ ký số tạm dừng hoặc thu hồi chứng thư số của thuê bao trong các trường hợp quy định tại Điều 15 Thông tư này.

Điều 8. Quyền và nghĩa vụ của thuê bao

1. Được cung cấp thông tin hướng dẫn về trình tự, thủ tục cấp phát, quản lý và sử dụng chứng thư số.

2. Thông qua tổ chức quản lý thuê bao của mình để đề nghị cấp, gia hạn, tạm dừng, khôi phục, thu hồi chứng thư số hoặc thay đổi cặp khóa.

3. Thuê bao có thể trực tiếp gửi văn bản đề nghị Tổ chức cung cấp dịch vụ chữ ký số tạm dừng chứng thư số của mình và phải chịu trách nhiệm trước pháp luật về đề nghị đó.

4. Sử dụng chứng thư số đúng mục đích đã đăng ký.

5. Bảo quản và sử dụng khóa bí mật, các dữ liệu trong thiết bị lưu giữ khóa bí mật theo chế độ "Mật".

6. Thông báo kịp thời cho Tổ chức cung cấp dịch vụ chữ ký số và tổ chức quản lý thuê bao của mình trong trường hợp phát hiện hoặc nghi ngờ chứng thư số, khóa bí mật không còn an toàn.

7. Tuân thủ các quy định khác của pháp luật về quản lý và sử dụng chứng thư số.

Điều 9. Nghĩa vụ của người nhận

1. Trước khi chấp nhận chữ ký số của người ký, người nhận phải kiểm tra những thông tin sau:

a) Hiệu lực, phạm vi sử dụng, giới hạn trách nhiệm chứng thư số của người ký và chữ ký số của Tổ chức cung cấp dịch vụ chữ ký số;

b) Chữ ký số phải được tạo bởi khóa bí mật ứng với khóa công khai trên chứng thư số của người ký.

2. Người nhận phải chịu mọi thiệt hại xảy ra trong trường hợp sau:

a) Không tuân thủ các quy định tại khoản 1 Điều này;

b) Đã biết hoặc được thông báo về sự không còn tin cậy của chứng thư số và khóa bí mật của người ký.

Chương III. DỊCH VỤ CHỨNG THỰC CHỮ KÝ SỐ

Điều 10. Đăng ký sử dụng dịch vụ chứng thực chữ ký số

1. Tổ chức, cá nhân tham gia sử dụng dịch vụ chứng thực chữ ký số của Bộ Công thương đăng ký một trong các thủ tục sau:

a) Cấp chứng thư số (quy định tại Điều 12 của Thông tư này);

b) Gia hạn chứng thư số (quy định tại Điều 13 của Thông tư này);

c) Thay đổi cặp khóa (quy định tại Điều 14 của Thông tư này);

d) Tạm dừng, thu hồi chứng thư số (quy định tại Điều 15 của Thông tư này);

đ) Khôi phục chứng thư số (quy định tại Điều 16 của Thông tư này).

2. Tổ chức, cá nhân có thể lựa chọn đăng ký qua mạng Internet tại địa chỉ http://www.vsign.vn hoặc đăng ký tại Trụ sở của Bộ Công thương - Cục Thương mại điện tử và Công nghệ thông tin, 25 Ngô Quyền, Hoàn Kiếm, Hà Nội.

Điều 11. Trình tự đăng ký sử dụng dịch vụ chứng thực chữ ký số qua mạng Internet

1. Tổ chức, cá nhân phải khai báo các thông tin vào phần mềm do Bộ Công thương cung cấp và gửi dữ liệu điện tử về Bộ Công thương. Hồ sơ nộp qua mạng Internet bao gồm:

a) Bản khai điện tử yêu cầu đăng ký sử dụng dịch vụ chứng thực chữ ký số của tổ chức, cá nhân;

b) Bản scan từ bản gốc quyết định thành lập của tổ chức quản lý thuê bao đối với hồ sơ đề nghị cấp chứng thư số lần đầu (không áp dụng đối với các đơn vị thuộc Bộ Công thương).

2. Các cán bộ tiếp nhận hồ sơ và tiến hành xem xét thông tin khai báo qua mạng Internet và thông báo kết quả kiểm tra qua mạng Internet về cho các tổ chức, cá nhân. Kết quả kiểm tra có thể thuộc một trong hai trường hợp sau:

a) Đồng ý qua mạng Internet trong trường hợp các thông tin khai báo qua mạng Internet phù hợp và hợp lệ;

b) Đề nghị tổ chức, cá nhân sửa đổi, bổ sung thông tin.

3. Đối với trường hợp yêu cầu sửa đổi, bổ sung thông tin, tổ chức, cá nhân tiến hành sửa đổi, bổ sung thông tin theo yêu cầu của tổ chức cấp và truyền dữ liệu khai báo này qua mạng Internet về tổ chức cấp để kiểm tra lại cho đến khi các thông tin phù hợp với yêu cầu của tổ chức cấp.

4. Sau khi nhận được thông báo chấp nhận của tổ chức cấp về việc thông tin hồ sơ khai báo qua mạng Internet đã đầy đủ, hợp lệ, đơn vị chịu trách nhiệm cung cấp dịch vụ chứng thực chữ ký số sẽ tiến hành cung cấp dịch vụ theo yêu cầu. Kết quả sẽ được trả về qua đường bưu điện hoặc trực tiếp tại trụ sở của Bộ Công thương.

Điều 12. Cấp chứng thư số

1. Điều kiện đề nghị cấp chứng thư số: Tổ chức, cá nhân đề nghị cấp chứng thư số phải thỏa mãn các điều kiện sau:

a) Điều kiện chung:

- Thuộc đối tượng theo quy định tại Điều 2 Thông tư này;

- Chấp thuận tuân thủ các quy định đối với thuê bao tại Thông tư này.

b) Điều kiện bổ sung đối với các đối tượng quy định tại khoản 2 Điều 2:

- Là doanh nghiệp được thành lập theo pháp luật Việt Nam;

- Có khả năng trang bị các thiết bị kỹ thuật, tổ chức và duy trì hoạt động phù hợp với hệ thống giao dịch điện tử của Bộ Công thương;

- Người đại diện theo pháp luật hiểu biết pháp luật về chữ ký số và dịch vụ chứng thực chữ ký số.

2. Hồ sơ đề nghị cấp chứng thư số:

Tổ chức, cá nhân đề nghị cấp chứng thư số có thể khai báo trực tuyến qua mạng Internet hoặc nộp tại trụ sở của Bộ Công thương (trực tiếp hoặc qua đường bưu điện). Trong trường hợp tổ chức, cá nhân lựa

chọn nộp hồ sơ qua mạng Internet sẽ thực hiện theo quy định tại Điều 11 của Thông tư này.

Trong trường hợp tổ chức, cá nhân lựa chọn nộp hồ sơ trực tiếp tại trụ sở Bộ Công thương, hồ sơ đề nghị cấp bao gồm:

a) Giấy đề nghị cấp chứng thư số của tổ chức, cá nhân, có xác nhận của tổ chức quản lý thuê bao;

b) Bản sao hợp lệ quyết định thành lập của tổ chức quản lý thuê bao đối với hồ sơ đề nghị cấp chứng thư số lần đầu (không áp dụng đối với các đơn vị thuộc Bộ Công thương).

3. Trong thời hạn không quá 05 (năm) ngày làm việc, kể từ ngày nhận được hồ sơ đề nghị cấp chứng thư số hợp lệ, tổ chức cung cấp dịch vụ chữ ký số có trách nhiệm kiểm tra, cấp chứng thư số cho thuê bao nếu đủ điều kiện hoặc có văn bản từ chối trong đó nêu rõ lý do từ chối nếu không đủ điều kiện cấp chứng thư số.

Điều 13. Gia hạn chứng thư số

1. Thủ tục gia hạn chứng thư số:

a) Chứng thư số được đề nghị gia hạn phải đảm bảo còn thời hạn sử dụng ít nhất là 30 ngày;

b) Tổ chức, cá nhân gia hạn chứng thư số có thể khai báo trực tuyến qua mạng Internet hoặc nộp tại trụ sở Bộ Công thương (trực tiếp hoặc qua đường bưu điện) giấy đề nghị gia hạn chứng thư số của thuê bao, có xác nhận của tổ chức quản lý thuê bao;

c) Mỗi chứng thư số được gia hạn không quá 03 (ba) lần, thời gian gia hạn cho mỗi lần không quá 01 (một) năm.

2. Thời hạn xử lý hồ sơ gia hạn chứng thư số:

Trong thời hạn không quá 05 (năm) ngày làm việc, kể từ ngày nhận được hồ sơ đề nghị gia hạn chứng thư số hợp lệ, Tổ chức cung cấp dịch vụ chữ ký số có trách nhiệm kiểm tra, gia hạn chứng thư số

cho thuê bao nếu đủ điều kiện hoặc có văn bản từ chối trong đó nêu rõ lý do từ chối nếu không đủ điều kiện gia hạn chứng thư số.

Điều 14. Thay đổi cặp khóa

1. Điều kiện thay đổi cặp khóa:

a) Có yêu cầu thay đổi cặp khóa của thuê bao và phải đảm bảo thời hạn sử dụng còn lại của chứng thư số ít nhất là 30 (ba mươi) ngày;

b) Tổ chức, cá nhân muốn thay đổi cặp khóa có thể khai báo trực tuyến qua mạng Internet hoặc nộp tại trụ sở Bộ Công thương (trực tiếp hoặc qua đường bưu điện) giấy đề nghị thay đổi cặp khóa của thuê bao, có xác nhận của tổ chức quản lý thuê bao.

2. Thay đổi cặp khóa được tiến hành như sau:

a) Đảm bảo kênh thông tin tiếp nhận yêu cầu thay đổi cặp khóa hoạt động 24 (hai mươi tư) giờ trong ngày và 7 (bảy) ngày trong tuần;

b) Trong thời hạn không quá 05 (năm) ngày làm việc, kể từ ngày nhận được hồ sơ đề nghị thay đổi khóa hợp lệ, Tổ chức cung cấp dịch vụ chữ ký số kiểm tra, thay đổi cặp khóa cho thuê bao;

c) Lưu trữ thông tin liên quan đến hoạt động thay đổi cặp khóa trong thời gian ít nhất 05 năm, kể từ thời điểm thay đổi.

Điều 15. Tạm dừng, thu hồi chứng thư số

1. Chứng thư số của thuê bao bị tạm dừng trong các trường hợp sau:

a) Tổ chức, cá nhân muốn tạm dừng chứng thư số có thể khai báo trực tuyến qua mạng Internet hoặc nộp tại trụ sở Bộ Công thương (trực tiếp hoặc qua đường bưu điện) văn bản yêu cầu từ thuê bao, có xác nhận của tổ chức quản lý thuê bao trong các trường hợp: khóa bí mật bị lộ hoặc nghi bị lộ; thiết bị lưu giữ khóa bí mật bị thất lạc, bị sao chép hoặc các trường hợp mất an toàn khác;

b) Theo yêu cầu bằng văn bản từ các cơ quan nhà nước có thẩm quyền;

c) Theo yêu cầu bằng văn bản từ tổ chức quản lý thuê bao;

d) Tổ chức cung cấp dịch vụ chữ ký số có đủ căn cứ xác định thuê bao vi phạm các quy định tại Thông tư này;

đ) Tổ chức cung cấp dịch vụ chữ ký số phát hiện ra bất cứ sai sót, sự cố nào có thể ảnh hưởng đến quyền lợi của thuê bao hoặc an ninh, an toàn của hệ thống cung cấp dịch vụ chứng thực chữ ký số.

2. Chứng thư số của thuê bao bị thu hồi trong các trường hợp sau:

a) Chứng thư số hết hạn sử dụng;

b) Theo yêu cầu bằng văn bản từ các cơ quan nhà nước có thẩm quyền;

c) Tổ chức, cá nhân muốn thu hồi chứng thư số có thể khai báo trực tuyến qua mạng Internet hoặc nộp tại trụ sở Bộ Công thương (trực tiếp hoặc qua đường bưu điện) văn bản yêu cầu từ thuê bao, có xác nhận của tổ chức quản lý thuê bao;

d) Theo yêu cầu bằng văn bản của tổ chức quản lý thuê bao;

đ) Tổ chức quản lý thuê bao, thuê bao bị giải thể hoặc phá sản theo quy định của pháp luật;

e) Có đủ căn cứ xác định thuê bao vi phạm các quy định về quản lý, sử dụng khóa bí mật và thiết bị lưu giữ khóa bí mật tại Thông tư này;

g) Thời gian tạm dừng chứng thư số tối đa là 06 (sáu) tháng.

3. Tổ chức cung cấp dịch vụ chữ ký số phải đảm bảo các yêu cầu sau:

a) Đảm bảo kênh thông tin tiếp nhận yêu cầu tạm dừng, thu hồi chứng thư số hoạt động 24 (hai mươi tư) giờ trong ngày và 07 (bảy) ngày trong tuần;

b) Lưu trữ thông tin liên quan đến hoạt động tạm dừng hoặc thu hồi chứng thư số trong thời gian ít nhất 05 (năm) năm kể từ thời điểm chứng thư số bị tạm dừng hoặc thu hồi;

c) Khi nhận được hồ sơ yêu cầu tạm dừng hoặc thu hồi chứng thư số của tổ chức, cá nhân hoặc khi có đủ căn cứ tạm dừng, thu hồi

chứng thư số, Tổ chức cung cấp dịch vụ chữ ký số phải tiến hành tạm dừng hoặc thu hồi chứng thư số trong thời hạn không quá 05 (năm) ngày làm việc.

Điều 16. Khôi phục chứng thư số

1. Chứng thư số khôi phục trong các trường hợp sau:

a) Theo yêu cầu bằng văn bản từ phía các cơ quan Nhà nước có thẩm quyền;

b) Tổ chức, cá nhân muốn khôi phục chứng thư số có thể khai báo trực tuyến qua mạng Internet hoặc nộp tại trụ sở Bộ Công thương (trực tiếp hoặc qua đường bưu điện) văn bản yêu cầu từ thuê bao, có kèm theo xác nhận của tổ chức quản lý thuê bao, trong trường hợp thuê bao, tổ chức quản lý thuê bao đã đề nghị tạm dừng chứng thư số trước đó;

c) Thời gian tạm dừng chứng thư số theo đề nghị tạm dừng đã hết;

d) Chứng thư số bị tạm dừng theo quy định tại điểm đ khoản 1 Điều 15 Thông tư này và những sai sót, sự cố đó đã được khắc phục.

2. Trong thời hạn không quá 05 (năm) ngày làm việc, kể từ ngày nhận được hồ sơ đề nghị khôi phục chứng thư số hợp lệ, Tổ chức cung cấp dịch vụ chữ ký số có trách nhiệm kiểm tra, khôi phục chứng thư số cho thuê bao.

Chương IV. ĐIỀU KHOẢN THI HÀNH

Điều 17. Xử lý vi phạm, khiếu nại và giải quyết tranh chấp

Việc xử lý vi phạm, khiếu nại và giải quyết tranh chấp liên quan đến việc thực hiện Thông tư này được thực hiện theo quy định của Nghị định chữ ký số và các quy định khác của pháp luật có liên quan.

Điều 18. Trách nhiệm thi hành

1. Cục Thương mại điện tử và Công nghệ thông tin có trách nhiệm:

a) Hướng dẫn, theo dõi và kiểm tra việc chấp hành Thông tư này của các đơn vị thuộc Bộ Công thương và các tổ chức khác có sử dụng dịch vụ chứng thực chữ ký số của Bộ Công thương;

b) Đảm bảo sự hoạt động ổn định, an toàn, liên tục của hệ thống chữ ký số, nghiên cứu và triển khai các công nghệ chữ ký số tiên tiến, phù hợp với hoạt động của Bộ Công thương.

2. Thanh tra Bộ Công thương có trách nhiệm phối hợp với Cục Thương mại điện tử và Công nghệ thông tin kiểm tra việc thực hiện Thông tư này.

3. Thủ trưởng các đơn vị thuộc Bộ Công thương và thủ trưởng các tổ chức khác có sử dụng dịch vụ chứng thực chữ ký số của Bộ Công thương có trách nhiệm tổ chức triển khai và kiểm tra việc thực hiện tại đơn vị mình theo đúng các quy định của Thông tư này.

Điều 19. Thông tư này có hiệu lực kể từ ngày 15 tháng 5 năm 2011 và thay thế Quyết định số 40/2008/QĐ-BCT ngày 31 tháng 10 năm 2008 về việc ban hành Thông tư quản lý, sử dụng chữ ký số, chứng thư số và dịch vụ chứng thực chữ ký số của Bộ Công thương.

Điều 20. Chánh Văn phòng Bộ, Cục trưởng Cục Thương mại điện tử và Công nghệ thông tin, Thủ trưởng các đơn vị trực thuộc Bộ Công thương và các tổ chức, cá nhân có liên quan chịu trách nhiệm thi hành Thông tư này./.

K/T. BỘ TRƯỞNG
THỨ TRƯỞNG

(Đã ký)

Hồ Thị Kim Thoa

THUẬT NGỮ VIẾT TẮT

Thuật ngữ	Tiếng Anh	Tiếng Việt
AEA	Advanced Encryption Algorithm	Thuật toán mã hóa tiên tiến
AS	Authentication Server	Máy chủ xác thực
CA	Certification Authority	Cơ quan chứng thực điện tử
CDC	Clear Data Channel	Xóa kênh dữ liệu
CRL	Certificate Revocation List	Danh sách các chứng thực bị thu hồi
DC	Differential Cryptanalysis	Phá mã vi sai
DEA	Data Encryption Algorithm	Thuật toán mã hóa dữ liệu
DES	Data Encryption Standard	Tiêu chuẩn mã hóa dữ liệu
DOS	Denial of Service	Tấn công từ chối dịch vụ
ECC	Elliptic Curve Cryptography	Mật mã đường cong elliptic
ESP	Encapsulating Security Payloads	Khối đóng gói an toàn
FIPS	Federal Information Processing Standard	Tiêu chuẩn xử lý thông tin Liên bang Hoa Kỳ
FTP	File Transport Protocol	Giao thức truyền tệp
FTPS	File Transfer Protocol Secure	Giao thức truyền tệp có bảo mật
HS	Hash Function	Hàm băm
HTTP	Hyper Text Transpot Protocol	Giao thức truyền tải siêu văn bản
HTTPS	Hypertext Transfer Protocol Secure	Giao thức truyền thông siêu văn bản có bảo mật
IDEA	International Data Encryption Algorithm	Thuật toán mã hóa dữ liệu quốc tế
IMAP	Internet Messaging Access Protocol	Giao thức truy nhập bản tin Internet

IPsec	Internet Protocol Security	Giao thức Internet an toàn
KDC	Key Distribution Center	Trung tâm phân phối khóa
LC	Linear Cryptanalysis	Phá mã tuyến tính
MIME	Multipurpose Internet Mail Extension	Giao thức mở rộng thư điện tử đa phương tiện trên Internet
MIT	Massachusett Institute of Technology	Viện Công nghệ Massachusett
NAT	Network Address Translation	Thay đổi địa chỉ mạng
NIST	National Institute of Standards and Technolgy	Viện Quốc gia về Tiêu chuẩn và Công nghệ
NLSP	Network Layer Security Protocol	Giao thức an ninh tầng mạng
NSA	National Security Agency	Cơ quan an ninh quốc gia
OCSP	Online Certificate Status Protocol	Giao thức trạng thái chứng thư trực tuyến
OI	Order Information	Thông tin mua hàng
PAKE	Password-authenticated key agreement	Thỏa thuận khóa xác thực mật khẩu
PCT	Private Communication Technology	Công nghệ truyền thông riêng tư
PI	Payment Information	Thông tin trả tiền
PKI	Public Key Infrastructure	Hạ tầng khóa công khai
RFC	Request for Comments	Bản phác thảo
S/MIME	Secure/Multipurpose Internet Mail Extension	Giao thức mở rộng thư điện tử đa phương tiện trên Internet - có bảo mật
SA	Security Associations	Tổ hợp an ninh
SADB	Security Association Database	Cơ sở dữ liệu của tổ hợp an ninh
SCC	Secure Command Channel	Kênh điều khiển an toàn
SFTP	SSH File Transfer Protocol	Giao thức truyền tệp bao vỏ sò
SKC	Secret Key Cryptography	Mã hóa với khóa bí mật
SPI	Security Parameter Index	Chỉ số tham số an ninh

SS	Service Server	Máy chủ cung cấp dịch vụ
SSH	Secure Shell Protocol	Giao thức vỏ sò bảo mật
SSL	Secure Socket Layer	Tầng đệm bảo mật
TLS	Transpot Layer Security	An ninh lớp giao vận
VPN	Virtual Private Network	Mạng riêng ảo

TÀI LIỆU THAM KHẢO

[1] Thái Thanh Tùng, *Giáo trình An ninh mạng và bảo mật dữ liệu*, Đại học Mở Hà Nội, 2006.

[2] Thái Thanh Sơn - Thái Thanh Tùng, *Thương mại điện tử,* NXB Thông tin và Truyền thông, 2011.

[3] Thái Thanh Sơn, *Đại số học*, NXB Đại học Quốc gia Hà Nội, 2004.

[4] Nguyễn Đức Nghĩa - Nguyễn Tô Thành, *Toán rời rạc*, NXB Đại học Quốc gia Hà Nội, 2004.

[5] Phan Đinh Dieu, Le Cong Thanh, Le Tuan Hoa, *Average Polynomial Time Complexity of Some NP-Complete Problems,* Theor. Comput. Sci. 46(3): 219-327 (1986)

[6] E. Biham & A. Shamir, *Differential cryptanalysis of DES-like cryptosystems, Journal of Cryptology,* Springer-Verlag, 1991.

[7] Alfred J. Menezes, Paul Van Oorschot, Scott A. Vanstone, *Handbook of applied cryptography,* CRC Press.1996.

[8] H.X. Mel & Doris Baker, *Cryptography decrypted,* Addison-Wesley 2003.

[9] Douglas Robert Stinton, *Cryptography: Theory and Practice,* Chapman & Hall/CRC, 2006.

[10] Gary C. Kesler, *An Overview of Cryptography*, Edition of Handbook on Local Area Networks, 2010.

[11] www.barcodesinc.com/articles/cryptography2.htm

[12] www.cryptography.com/

MỤC LỤC

GIÁO TRÌNH MẬT MÃ HỌC VÀ HỆ THỐNG THÔNG TIN AN TOÀN

Chịu trách nhiệm xuất bản
NGUYỄN THỊ THU HÀ

Biên tập:	NGÔ MỸ HẠNH
	TRỊNH THU CHÂU
Trình bày sách:	BÙI NGỌC BẢO
Sửa bản in:	TRỊNH THU CHÂU
Thiết kế bìa:	TRẦN HỒNG MINH

NHÀ XUẤT BẢN THÔNG TIN VÀ TRUYỀN THÔNG

Trụ sở: Số 9, Ngõ 90, Phố Ngụy Như Kon Tum, Quận Thanh Xuân, TP. Hà Nội
ĐT Biên tập: 04.35772143 ĐT Phát hành: 04.35772138
E-mail: nxb.tttt@mic.gov.vn Fax: 04.35772194, 04.35779858
Website: www.nxbthongtintruyenthong .vn
Chi nhánh TP. Hồ Chí Minh: 8A đường D2, P25, Quận Bình Thạnh, TP. Hồ Chí Minh
Điện thoại: 08.35127750, 08.35127751 Fax: 08.35127751
E-mail: cnsg.nxbtttt@ mic.gov.vn
Chi nhánh TP. Đà Nẵng: 42 Trần Quốc Toản, Quận Hải Châu, TP. Đà Nẵng
Điện thoại: 0511.3897467 Fax: 0511.3843359
E-mail: cndn.nxbtttt@ mic.gov.vn

In 1000 bản, khổ 16x24cm tại Công ty In Hải Nam
Số đăng ký kế hoạch xuất bản: 579-2011/CXB/15-166/TTTT
Số quyết định xuất bản: 176/QĐ-NXB TTTT ngày 20 tháng 7 năm 2011
In xong và nộp lưu chiểu tháng 7 năm 2011